

**blender**

**art**  
MAGAZINE

Blender learning made easy

**2<sup>nd</sup>**

**Anniversary  
Issue!**

**Making a 'Realistic' Underwater Rift**

**Creating a 'Tree Character'**

**Using 'Auto Masonry' Script**

**Video Editing in Blender**

**Making of - Andy Tear Liquid Tubes**

**Making of - Glass Girl**

**Meet the Blenderart Team**

**Fantasy  
special!**

**EDITOR**Gaurav Nawani [gaurav@blenderart.org](mailto:gaurav@blenderart.org)**MANAGING EDITOR**Sandra Gilbert [sandra@blenderart.org](mailto:sandra@blenderart.org)**WEBSITE**Nam Pham [nam@blenderart.org](mailto:nam@blenderart.org)**DESIGNER**

Gaurav, Sandra, Alex

**PROOFER**

Kevin Braun  
Phillip Ryals  
Bruce Westfall  
Derek Marsh  
Joshua Leung  
Lynda Schemansky  
Eric Pranausk  
Noah Summers  
Joshua Scotton  
Mark Warren  
Wade Bick  
Patrick O'Donnell  
Brian C. Treacy  
Scott Hill  
Henriel Veldtmann

**WRITERS**

Abhisek Gupta  
Dan Tabonet  
Denis  
Juan Pablo Bouza  
Migero  
Mark (dwarf)  
Nick (Rusty246)  
Paul Spooner (dudecon)  
Sandra Gilbert  
Yann Caron (CyaNn)

**COVER ART**

Zoltan Miklosi 'Guard of the Golden Horn'

# CONTENTS

2

Making a Realistic Underwater Rift

[7](#)

Deep Dream's Caustics

[10](#)

Creating a 'Tree Character'

[14](#)

Using Auto Masonry Script

[19](#)

Video Editing in Blender

[23](#)

Making of - Andy Tear Liquid Tubes

[27](#)

Making of - Glass Girl

[31](#)

Making of - Tripeace Animated

[36](#)

Making of - The Dance of the Bashful Dwarf

[38](#)

Meet the Blenderart Team

[51](#)



**Sandra Gilbert**  
Managing Editor

Welcome to our second anniversary issue of Blenderart! Looking back, it is hard to believe that two years have actually passed since we released our first issue. Time sure flies, and both Blender and Blenderart magazine have grown considerably in that time. We have covered a variety of topics and showcased a number of wonderful Blender projects by some of the most talented artists in our community.

And just like Blender itself, we will continue to grow and showcase all the best and brightest of our community to inspire and teach those of us still learning our way around the wonderful world of Blender.

I would like to take this opportunity to thank all of our volunteers, past and present, that have made Blenderart what it is today. I would also like to welcome all of our new proofreaders to our little staff of dedicated volunteers. They are a welcome addition. You can check out their biographies in our "Meet the Staff" article.

Now let's talk "Fantasy". Fantasy art is some of the most beautiful, as well as inspiring, art that you can view. From the traditional fairy, goblin and dragon to the fantastic dreamy looks into an artist's mind, there is a style that appeals to just about everyone. And while there are some loose rules about how some

characters are supposed to look, for the most part anything goes. This gives the artist a great deal of freedom when creating, and create they do!

For this issue we have rounded up some fascinating looks into how a number of artists go about creating their fantasies. We also show you how to create a tree character using Multires and Sculpt mode, and even take a look at a couple of games created in Blender's game engine.

So sit back and soak it all in. Then go model your own fantasy!

Happy Blending!  
[sandra@blenderart.org](mailto:sandra@blenderart.org)



*One of the best places to get inspired is the toy aisle of your favourite department store!*

Fantasy creatures/characters come in a wide array of styles and types. So many in fact, that it can be hard to decide what you want to model/create. It often helps to have a reference object or image to jump start the process. Even if your final model only slightly resembles the reference object, it can at least get you started and your imagination can take over at any time.

One of the best places to get inspired is the toy aisle of your favorite department store. In fact, that is my all-time favorite place to wander around. Being surrounded by toys puts me in the proper frame of mind and takes me back to a time when everything was magical.

Of course, the process is helped by the fact that fantasy genre toys are very popular right now. You can find dragons, fairies, and castles in a variety of detail levels. Depending on the manufacturer/brand, you can find everything from a simple toony look to hyper realism.

While I am not promoting any one toy manufacturer over another, I thought I would share some toys that I think are inspiring or at the very least make a great reference object to model from.

- Megabloks Dragons series: The sets, of course, feature nicely detailed dragons, with some sets also containing castles and a variety of medieval battle machines.
- Barbie Fairytoria: Yeah, I know - Barbie??? Nevertheless, there are some nice fairy dolls in a variety of styles and sizes.
- Bratz Fairies: Yeah, another fashion doll, though not your typical fairy look, which is what appealed to me when I saw them.

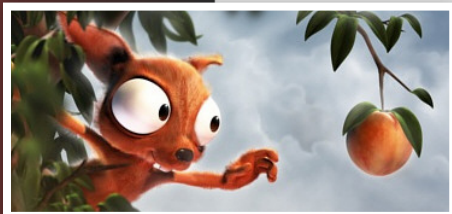
- Schleich Dragons and Knights: Highly detailed figures of dragons, knights, and horses, as well as castles, and medieval battle machines.
- Fisher-Price Imaginext Adventures: These are for the younger child and as such are more whimsical and colorful, yet still contain a lot of detail.
- Magna Morph Dragon Magnetic Building System: Nicely detailed dragon parts that you can mix and match to create your own unique dragon look.

Also, don't forget to look down the action figure aisle. That is where you most often find toys that have been marketed from major movies, so the selection changes as often as hit movies come out. Thus, constantly giving you a changing selection of nicely detailed figures to choose from.

Okay, so now you have the actual reference object (read that as the toy you bought not only to model but also to play with when no one is looking'). Having it in your hand is the ideal scenario, but it can also become an expensive one as you continue to discover inspirational objects in the toy department. Not to worry, most toy manufacturers post images from different angles on their websites that generally are useful and detailed enough for modeling purposes. You could also take a camera to the store with you and take your own photos, or you can raid your child's (or niece/ nephew/ younger sibling/ next door neighbours kid's) toy box to see what inspiration it contains.

The important thing to remember is to always be on the lookout for objects that will inspire your creative process. You never know where you will find them. ●





## Project Peach

Project Peach is off to a great start. They have produced and posted their first animatic from their story board as well as a video showing Nathan rigging Rinky the squirrel. To see the latest progress on

Project Peach, check out their project site.

Project Peach is looking for some extra coders to help out in creating volumetric clouds. If you are interested, you can check out their requirements here.

## The ManCandy FAQ

The first in an Open Movie Workshop series is available for pre-order.

**The ManCandy FAQ** is a collection of animations and video tutorials for rigging and animating with ManCandy. The animations are in question and answer format (hence the title) and are intended to be a fun way of documenting the rig.

The DVD includes an HTML interface that contains some written tutorial/reference material, a collection of videos (Animations in 640x480 resolution, Tutorials in 1024x768 resolution), the ManCandy 2.0 blend file, the mystery character blend file, all the blend files for the animations on the DVDs, and of course some helper blend files for the tutorials.



## Material Node Cookbook

Blenderart Magazine is launching a special project, a Material Node Cookbook.

As Blender users become more familiar with the Material Node system, a lot of very cool Material Node setups are being created. Unfortunately, the majority of these setups are spread all over the net on various forums, threads and websites. BAM thinks it would be beneficial to the community to gather up as many of these wonderful setups (with the artists' permission of course) as we can find, add a short 'how to/description' of how and why it works and bundle it all in a pdf format 'Cookbook'.

It would allow node users to quickly get the look they need without re-inventing the wheel, so to speak. It would also provide a valuable learning tool for users that are still struggling with the whole Node system.

After giving it some thought, I am considering some of the following format options:

1. PDF chapters separated in logical categories for a smaller download size .
  - each category could be updated quickly and or a new pdf with additional nodes could be created.

## Material Node Cookbook ...

1. A flash card based system (or more appropriately recipes cards) with the node set-up image on one side of the card and the description/how to on the reverse side
2. Recipe cards could be added as need with 1st sets bundled according to category and future releases in one download that could be added to your previous sets.

Preliminary categories (to be adjusted later as set-ups are submitted)

1. Organics
  - Skin
  - Nature
  - Wood
  - Food
  - etc.
2. Inorganics
  - Building
  - Cloth
  - Dirt
  - Metal
  - Patterns
  - Glass
  - etc.
3. Special Effects
4. Compositing

We welcome any and all submissions, in fact the more that are submitted the better our cookbook will be.

To submit a Node set-up please follow these guidelines

1. Node set-up must be an original creation by the person submitting it. Please do not submit someone else's Node set-up, as then I will have to track down the original artist to obtain permission
2. Please send a blend file containing your Node set-up. I will be taking all the screen shots to maintain a consistent look/size for the cookbook
3. If you want to write the description/how-to for your Node set-up, feel free to do so, otherwise I will be happy to take care of that myself.

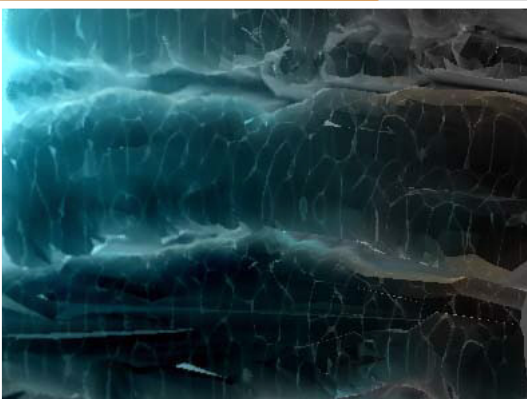
So dust off that amazing setup you created and send it in. The community will love you for it.

Send all submissions to [sandra@blenderart.org](mailto:sandra@blenderart.org).

Please put Node set-up in the subject line area.●

# 3D WORKSHOP: Making a 'Realistic' Underwater Rift

7



## Introduction

This tutorial is based on the works done for the 3d rendered film '[Deep dreams](#)' by the author Dan Tabornet.

**Step1:** First, in Blender, lets create a horizontal plane (go to top view [Numpad 7] and click Add>Mesh>Plane). Press the S-key to scale it to nearly fill the view and then left click. In edit mode, press the F9-key to subdivide it, say 5 times, then deselect all by pressing the A-key. Press the B-key twice to obtain the Brush Selection tool. Use the mouse wheel to adjust the size of the brush until it's about the size of a face, like in Figure1.

**Step2:** Using the left mouse button, draw lines randomly on the mesh, selecting what will be the hollow parts of the rift. Don't make them too large. You should obtain something like in Figure2 (Top view). Finish by right clicking -once only!

**Step3:** Pressing the numpad 1 or 3-keys (front and side view, respectively) will allow you to see your plane in a lateral view. Then, turning proportional editing off, move the newly selected vertices a little above the others, for instance at a distance of 5 to 10 times the size of an edge, like in Figure3.

**Step4:** Exit the Edit mode by pressing the Tab-key. In the Multires Panel, press Add Level once, then Apply Multires. (tutor001.blend), Figure4.

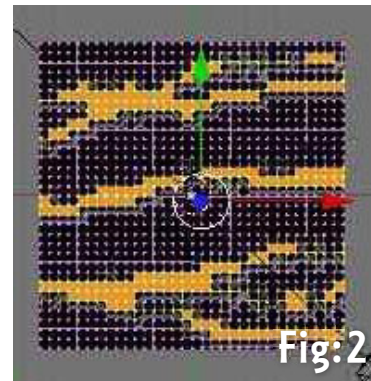


Fig:2

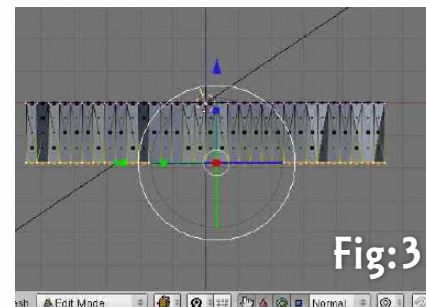


Fig:3

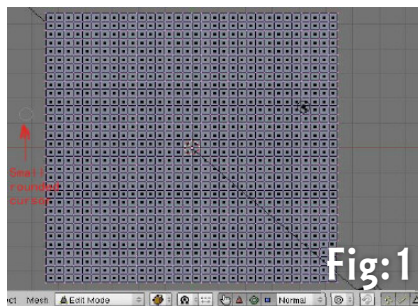


Fig:1

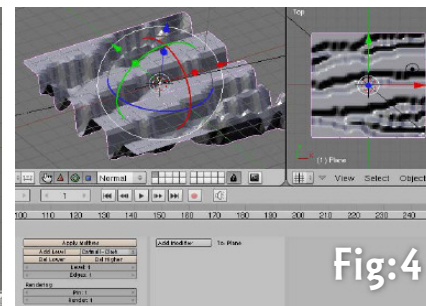


Fig:4

by Dan Tabornet

# 3D WORKSHOP: Making a 'Realistic' Underwater Rift

8

**Step5:** Return to Edit mode. To reduce the number of selected points to move let's press the Control-Menus keys (numpad subtraction sign "-"). You can now push these more deeply above the others again like in Figure5.

**Step6:** Now in object mode (Tab-key), in the Modifier panel add a Decimate modifier of a ratio like 0.3 and Apply. Do *not* go too low and alter the shape too much. (tutor002.blend) and Figure6.

Now toggle to Sculpt Mode. In Multires Panel add 1 Level. Don't apply for now. In the Sculpt Panels set Brush to Inflate, Shape to Add, Size to 50, and Strength to 25 with Symmetry off.

**Step7:** You can now use the middle mouse button place the perspective view (orthogonal for instance by pressing the numpad 5-key) to allow you to draw in the regions of the rift which are in the rounded borders to make these look more like Figure7. (tutor003.blend)

You can now Apply Multires when sculpting is done in different points of view, until you are satisfied. An Auto Smooth of less than 30 degrees in the Mesh panel applied with a Set Smooth in the Link and Materials Panel will make the rocks look more realistic.

In Vertex paint Mode, you can set a greeny-blue colour for vertex points, and the excellent Self shadow VCols script, faking ambient occlusion can be applied twice, to make the deeper sides be darker. If you try to paint in vertex mode, use it VERY sparingly to keep realism on your scene. Last notes on the materials:

The material used for the rift must then be set with the VCol Paint button ON to see the different vertex colour modifications made before. You also need to set the material Specularity very low: 0 would be good enough.

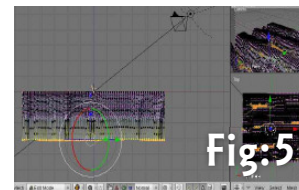


Fig:5

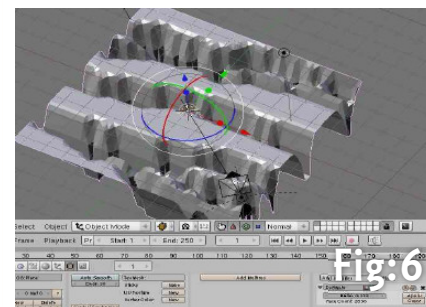


Fig:6

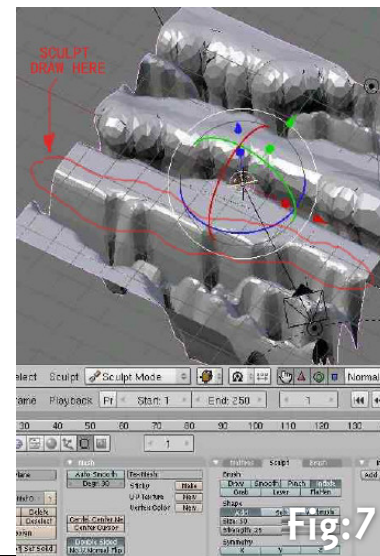


Fig:7

by Dan Tabornet

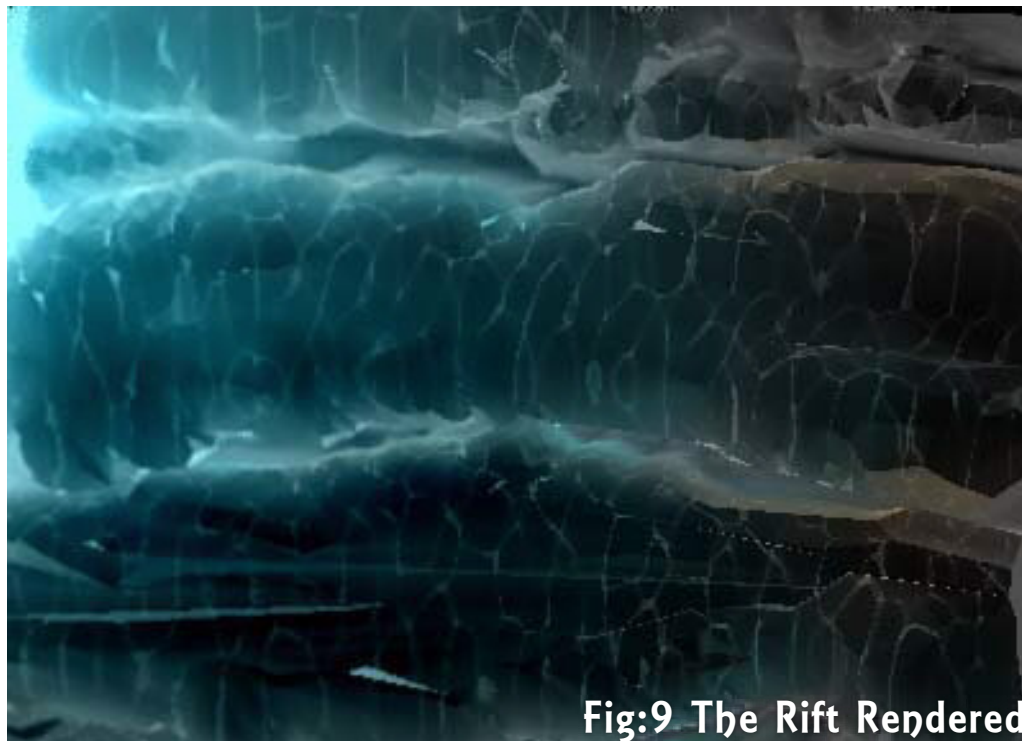
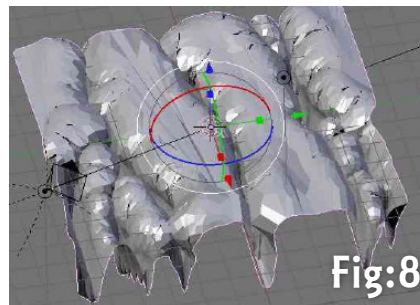


# 3D WORKSHOP: Making a 'Realistic' Underwater Rift

9

Here it is: the final render in Figure9.

If the lighting in your scene is like in the previous tutorial about caustics, you will get something that blend-fishes will be glad to swim in! ●



by Dan Tabonet



Deep Dreams

by Dan Tabornet

## Introduction

When I was an 8 year old kid, some of my young friends used to collect mysterious things in water ponds near my home. I'll always remember the day when, for the first time, I could see what was so exciting to them: thousands of very little, very shiny thunderbolt creatures in a so little flask of water they showed to my wide open eyes! It was so amazing that I saw myself at the size of one of them, swimming among them... and that was *the* dream! So,

since that day I have asked myself how to share this.

Many years passed before I saw one of the first 3D software packages in early 1993; it was Sculpt Animate4d on the Amiga platform. That was a kind of revelation - and a challenge: making others see my dream! But at this time modeling a single chess piece and making it turn on itself was a "piece" of work, since rendering *one* frame at 640x480 resolution could take half an hour!

However, when more sophisticated software came for my needs of modelling insects, I trained a lot with my dream still in mind. And believe me, that kind of modeling was much more complicated than making space vessels. Just have a look, for instance, at the images in the 12 volumes of the Larousse Encyclopedia of Insects and you will be convinced that imagination is nothing compared to

the real world, because it is in a real brain: yours! That was an artistic but very technical challenge as well. So if I had a word to future animators who hesitate to show their work, my motto to you is:

- Observe from nature
- Learn from nature
- Modestly imitate, and you will go where no one has gone before where dreams get real (appearance)...

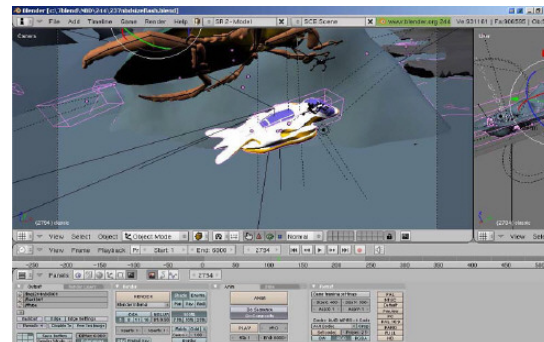
## Technical Details

Talking about all the "making of" the movie would take too long to explain in detail. Some examples of amount of work, for now: 14 months of perfecting the model more or less 250 times for a finally -compressed- .blend file of 73 Mb including 555 meshes, 906 595 polygons, 107 Mb of textures... you begin to see the challenge and the drastic choices I had to make for the final render, and that taught me precious things:

For instance, I now get a better flow of production because I lost a lot of time scaling and positioning the new meshes in the scene instead of creating them by the means of temporary reference boxes (simple cubes created where I projected to make new objects later) in the scene, exporting these boxes out of the scene and modelling into them the new objects separately, finally introducing the real new objects after wards into the scene.

Another precious hint I learned when I made my movie is that all objects that came out off the came

out off the camera field had to be hidden in layers by the means of layer keys, to make the render be possible on one machine in a "reasonable" time. For the same reason I had to turn raytracing off and use only shadow map lights. That increased the render speed a lot, but there was a serious problem to keep realism: for reflections of underwater scenes I had to find something to produce caustics. These are animated, random wavelets of light made by refraction on the water surface that you can see under it. This phenomenon is hard to produce in 3d, but very common in nature. If I wanted to recreate underwater ambiance, I couldn't do without it. So when I discovered the excellent free software, [Caustics Generator](#), that convinced me to fake caustics.



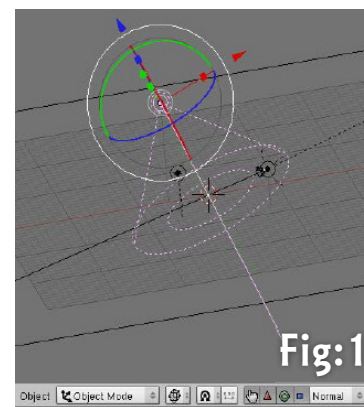
Now let's see how it works to make a relatively realistic underwater scene environment. In this little tutorial we assume basic knowledge of the Blender interface.

## First the scene lighting.

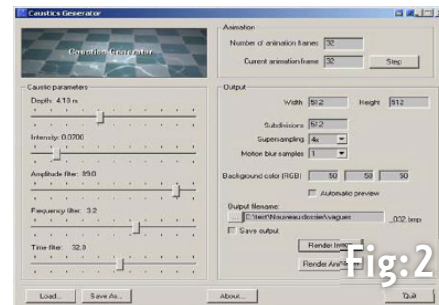
You need 3 lamps:

- "Projector lamp": First of all, add a spotlight in a clear 3d window in Blender, and turn it more or less downwards. Set the spot as a Buffered-shadow, then in a classic-halfway toggle, then the spotbias angle more than 0.5, to soften the border, and make it halo.
- "Soft lamp": Add another lamp (Hemi type) pointing behind the camera, and parented to it. Set its power to a high level, more or less 2. You'll see why we use this lamp later
- "Darkening lamp": Add Area lamp(s) set to negative power 0.5 or less, positioned upon the details to hide from view. This kind of lamp will darken unnecessary parts of the scene, hiding them from view.

To make the ground, add a plane in the top view and scale it until its borders are out the camera's field of view. Set the world colors to black. You should see something like this (figure1). (saustics001.blend)



Now it's time to prepare our animated caustics sequence of images or movie that we'll use as a texture for the projection lamp in Blender. So, first launch Caustics Generator and set the parameters like those in figure 2 (For now these settings would NOT be changed. Later on you will be able to rapidly test different other settings but if you check the automatic preview, be careful before this to disable antialiasing by setting Supersamples to 10 and Motionblur to 1, otherwise drawings would be drawn very slowly). Then, after setting the directory where you store the sequence of .bmp images, check Save Output and click Render Animation.



Here we are: A complete tileable cyclic animation of caustics, now fully useable in blender on the "projector lamp". But we should use it with care, to avoid some artifacts, because tiled textures can easily appear as artificial to our eyes. Let's return to Blender.

Our scene opened, let's select the projector lamp and add a texture to it. Next go to the Textures panel and set it like in figure 3. This indicates that the sequence used contains 32 frames (the parameter determined in CausticsGenerator), each of them used one by one and repeated cyclically in time forever. But at this point no control of the size of the pattern used is possible. We have to fix it.



A smart method is to put a plane under the ground and name it REF, for instance. The only thing to do is to enter Lamp buttons »Textures and Input, check the Object box, and enter the same name in this field: REF. Now, the mapping of that textured lamp will adapt to the size and position of the object named REF, like in figure4 (caustics002.blend).



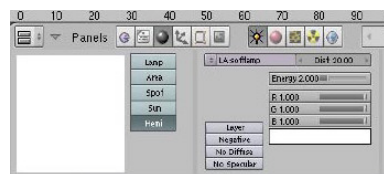


You should now be able to test rendering. After this, press F12.

If the patterns of the caustics are too large, you only need to resize the plane object you named REF in lower dimensions, and if they are too small make your REF object bigger. It's that simple! Now, with the REF plane selected, insert a loc key. Then go, say, 100 frames later and slightly move (horizontally only) the REF plane a little. Insert a second loc key on it. You have simulated the effect of the wind on the waves, making the caustics drift along with it.



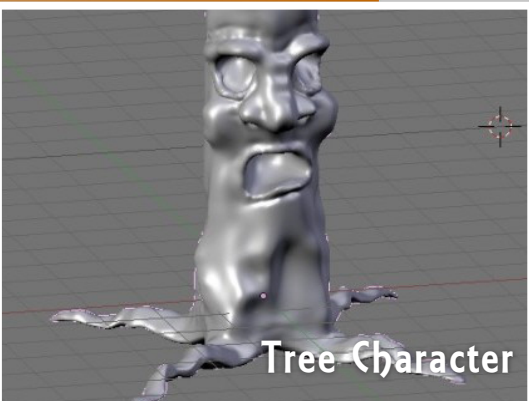
Let's push the Anim button and start testing... Good blending, and don't sink too much in it ;) ●



**Lamp Settings**



by Dan Tabor



by Sandra Gilbert

## Introduction

One of the things I love most about fantasy modeling is that anything is possible, up to and including tree characters. I love the idea of a tree character with features flowing naturally into the tree trunk and have often tried to model one. I have ended up with some rather interesting experiments, but none as interesting or as much fun to make as when I started using the Sculpt mode to create the face on the tree and give it some character. Although Halloween is already over, there is no harm in getting a head start on a spooky old tree for next year.

Sculpt mode allows for a great deal of natural/organic detailing that would be tedious and time consuming to model using traditional methods. It is also a lot of fun. The only real drawback is that the vertex count can quickly climb and bog down your machine if you are not careful.

**NOTE:** Do not be alarmed if your tree does not end up looking exactly like mine. Due to the nature of Sculpt mode which makes every model unique, I will be giving you a general road map to making a similar model, not an exact replica.

Before we launch into the wonders of Sculpt mode, we are going to model a very basic low-poly tree shape. This will be the base model for our higher

resolution model. So let's fire up Blender and get started.

## SCULPT TOOLS

**Draw** creates a smooth curve on the model following the brush; vertices are displaced in the direction of the average normal of the vertices contained within the brush. (hotkey: **D**)

**Smooth** As the name suggests, *Smooth* eliminates irregularities in the area of the mesh within the brush's influence. (hotkey: **S**)

**Pinch** pulls vertices towards the center of the brush. If *Sub* is active instead of *Add*, vertices are pushed away from the center of the brush. (hotkey: **P**)

**Inflate** is similar to *Draw*, except that vertices in *Inflate* mode are displaced in the direction of their own normals. (hotkey: **I**)

**Grab** is used to drag a group of vertices around. Unlike the other brushes, *Grab* does not modify different points as the brush is dragged across the model. Instead, *Grab* selects a group of vertices upon pressing and holding the mouse button, and pulls them to follow the mouse movement. The effect is similar to moving a group of vertices in Edit mode with proportional-editing enabled, except that *Grab* can make use of other Sculpt mode options (like textures and symmetry.) (hotkey: **G**) (continued on next page...)

## SCULPT TOOLS

**Layer** The *Layer* brush is similar to *Draw*, except that the height of the displacement layer is capped. This creates the appearance of a solid layer being drawn. This brush does not draw on top of itself; a single brush stroke intersects itself. Releasing the mouse button and starting a new stroke will reset the depth and paint on top of the previous stroke. (hotkey: L)

**Add and Sub:** *Add* causes the brush to pull an area of the model in the positive direction, *Sub* in the negative direction. (With the *Pinch* brush, *Add* pulls vertices inward and *Sub* pushes vertices outward.) *Interactive toggling of brush direction is done by holding down Shift. Alternatively, pressing the "V" key can be used to toggle it until it is toggled again.*

**Trunk** We will be creating a very basic trunk.

- We will start with a cube; use Spacebar > Add > Mesh > Cube. Select the top row of vertices and extrude (E key > Region) up 4 times. It might help to be in Wireframe mode while selecting top row of vertices (fig. 1). This will be the trunk of our tree.

**Main Branches** Next we will be creating a few main branches. For the scope of this tutorial, we are only going to make a few branches to keep the vertex count down, but feel free to add as many as you want.

- Select a side face at the top of the trunk (fig. 2), extrude (E key) 4-5 times, scaling (S key) each extrusion a little smaller than the last one. Go ahead and move (G key) and rotate (R key) the sections of the branch to create a pleasing profile (fig. 3).
- Go ahead and make another main branch on the other side using the same techniques as in the last step. You can make smaller branches by selecting faces on each branch and extruding, moving and scaling just as you did for the main branches (fig. 4).

**Tree Roots** Next we will create the roots. I experimented with various methods and this seemed to give the best results.

- Create a loop cut (Ctrl + R key) at the bottom of the trunk (fig. 5).
- Select the bottom row of vertices on the trunk and delete **only Edges**, leaving the four corner vertices in place (fig. 6).
- Select the vertices of the new row you created earlier (fig. 7) and subdivide them (W key > Subdivide).
- Now select the middle vertices in that row, making sure not to select any of the corner vertices (fig. 8).
- Move those 4 vertices down even with the vertices left from the bottom row, and fill in the missing faces by selecting three vertices at a time and pressing the F key (fig. 9).

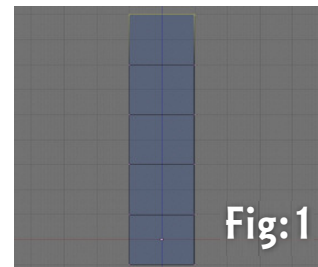


Fig:1

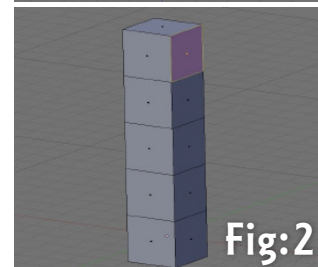


Fig:2

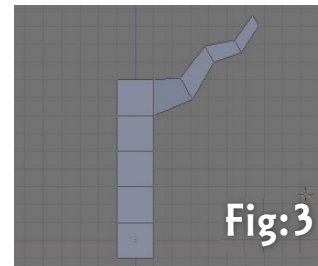


Fig:3

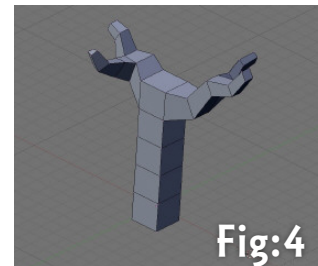


Fig:4

- Now using the same method, create faces to fill in the bottom of the trunk (fig. 10)

At this point you will notice that on each corner there are two triangle faces, which will be your roots. I will walk through one root and then you can just repeat the steps for the other three roots.

- Select a triangle face on both sides of a corner, extrude (E key) out 4 times, scaling each section smaller (S key). *Tip: when extruding press the middle mouse button to have free movement vs. constrained movement.*
- You can go ahead and move the sections up and down to make it look more interesting (fig. 11).
- Now, go make the other three roots and then we can start sculpting some 'character' into our character.



Fig:5

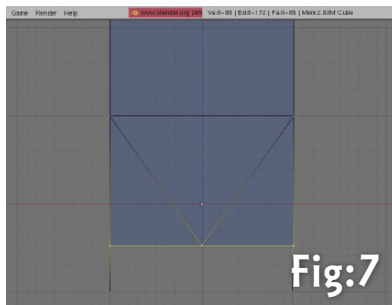


Fig:7

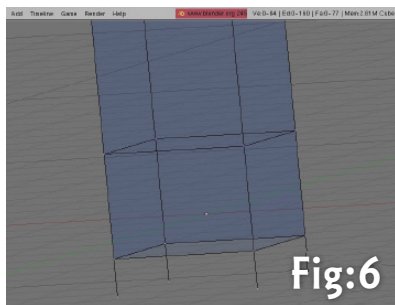


Fig:6

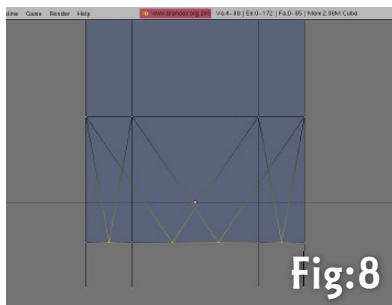


Fig:8

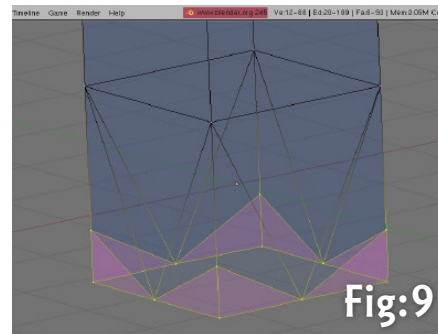


Fig:9

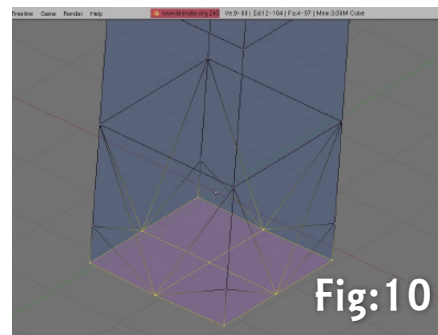


Fig:10

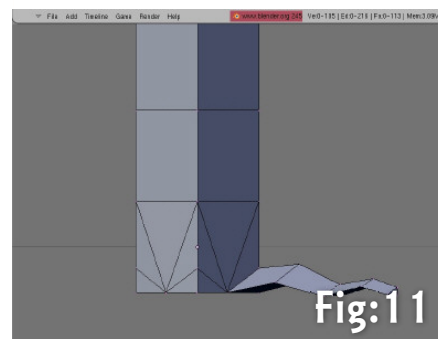


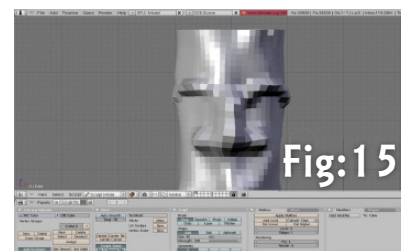
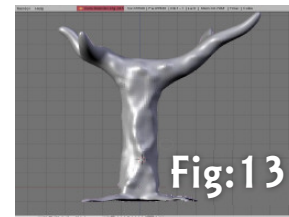
Fig:11



## Sculpting

So now we have our base model ready to sculpt. Granted it doesn't look like much yet, but it will give us a nice place to start doing some really cool things.

- First we need to be in Sculpt mode, and we need to add some Multires levels. I'm going to start at level 4 (fig. 12).
- Next up is creating a tree like surface. You know, the bumps and whorls that form as a tree grows? So with the Draw brush (D key) set to a size of 20 (F key) and strength of 100 (Shift + F key), start dragging the brush up and down the tree. Remember to rotate your view and draw over all of the tree, overlapping strokes if you want to. You can also hold the Shift key down while dragging the brush to create cracks/indents in the trunk surface (fig. 13).
- Add another Multires level, and now we will sculpt some eyes. On the Brush panel press the "X" toggle button under Symmetry to create mirrored features along the X axis. With the Draw brush set fairly small (15-20) draw a half circle on one side of your tree. If you've enabled symmetry, you should see the half circle appear on the other side as well (depending on the results of the previous step, your features might not be an exact mirror copy) (fig. 14).
- Increase the brush size to around 40 and draw a line down between the eye ridges; this will form the nose. Draw the nostril part of the nose to the side (small circle motion should do it) (fig. 15).
- Go back to the eye ridges we already formed, hold down the Shift key and draw under the ridges to create an eye socket. You can use the same settings from the previous step (fig. 16).
- Add another level of Multires. Now change the brush size to 15, and trace around the socket you just formed (fig. 17).
- Create nostrils by holding down the Shift key and drawing with a small brush size of about 10-15 (fig. 18).



# 3D WORKSHOP: Creating a 'Tree Character'

18

At this stage you have the beginnings of a face and it is up to your own creative vision and experimentation to finish him up. I will however, leave you with a few tips:

- Use the Inflate brush to exaggerate features (hold down the Shift key to shrink them down a little).
- The Smooth brush is your friend, it will even out any jagged marks/brush strokes.
- Use the Grab brush to pull out portions of your mesh, you could create smaller branches this way as well as give your tree more character.
- Only add levels (Multires) as you need them. They eat up a lot of memory.
- To speed up your work flow, hide parts of the mesh except the area you want to work on. Using Shift + Ctrl while dragging with the left mouse-button, you can define the area which will remain visible. The rest will be hidden from view and can be toggled back into view with Alt + H key.

Here is my tree character after a little more playing. I added a mouth using the Draw brush (and Shift + Draw). I also used the Inflate brush to exaggerate some features and the Smooth brush to soften a few areas. Now he is ready to texture, but that is a topic for a whole other tutorial in itself.

Once textures/materials are added, he is ready to star in the spooky scene of your choice.

I hope you had fun and were able to see the possibilities of Sculpt mode and Multires modeling •



Fig:16

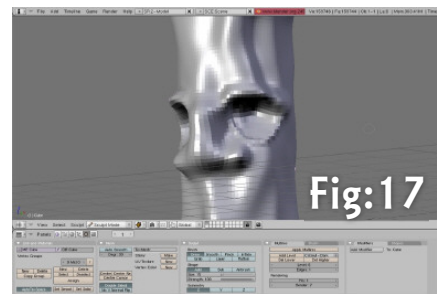


Fig:17



Fig:19

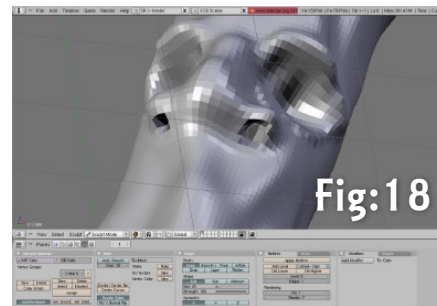


Fig:18

by Sandra Gilbert



## by Paul Spooner

AKA: Dudecon or Ziggy

Location: CA, USA

Interests: The varied forms of craftsmanship, design, music, imagination, and general Excellence.

Education: BA in Engineering: Mechanical concentration  
E-mail: [Dudecon@hotmail.com](mailto:Dudecon@hotmail.com)

### Introduction

I've always liked castles. When I started doing 3D modeling I used a tool called "Ray Dream Designer" and the first tutorial I did was making a castle. It was a wimpy castle with flat, poorly textured walls, no windows, etc. One thing that has always bugged me is that castle wall models tend to be textured planes, perhaps whitewashed with cornerstones modeled. Real castles have every stone showing, uneven rows and stone sizes, etc. The pure texture version

looks good from a distance, but you can't get close to them without the sham appearing. It's also hard to knock out a few blocks, or add siege damage. I've always dreamed of building a castle that would stand up to scrutiny, but the tools at hand were insufficient. I wasn't going to build it block by block!

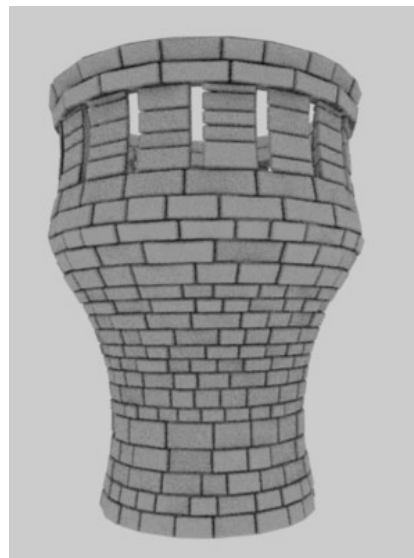
Anyway, I gave up on the dream for a while, moved on to newer software versions, and eventually switched to Blender. With the presence of the Python extension language my dream resurfaced. If the computer could do the grunt work, then all I'd have to do is the overall design and touch up work. Score! I looked around to see if anyone else had done this already, but I couldn't find anything similar to what I wanted. This confused me, it seemed like a simple problem to solve. More than six hundred lines of code later I discovered why it may not have been as simple as I expected.

What follows is my first major foray into python

scripting and the results. A section on how to use the script comes first, followed by a short history of the coding, and some things I learned in the process. I'm not fully satisfied with the script yet, but it is very usable and I hope will be useful to many of you.

### Using Auto Masonry

Let's jump straight into the fun stuff shall we? If you don't have it, download the current version of the script and place it in your scripts path folder (Normally "...Blender/.blender/scripts"). To start the script, open a "Scripts" window and select "Scripts»Wizards»Auto Masonry". You should see something like this.



It's the default GUI. If you press "Make this wall!" you should end up with something like in figure2

If you don't, you may need to install Python. If that doesn't work, I may need to fix the script. The current incarnation allows you to make walls with or without rectangular windows, doors, and crenellations. The script generates two mesh objects, a "wall" and a "grout".

*All of the buttons and options have explanatory pop-up text, so feel free to try it without reading this whole tedious section.*

If you're not feeling quite that bold, here is an explanation of all the options in the GUI.

- **"Make this wall!":** This button generates the wall which conforms to the options you have selected. If there is a curve selected, the wall length will be set to the curve length, the wall will deform to follow the curve, and have the curve as a parent. This means that all of the pos/rot/size of the curve will be applied to the wall as well. If the curve is closed the script will assume that it is a perfect circle. If the curve is not a circle, the wall probably won't be the right length.
- **"WALL":** These settings govern the overall size of the wall in blender units. The "Wall Height" does not include crenels (see "CRENELS", below).
- **"Straight Edges/Offset Edges":** Basically, if you want to make a closed loop wall set it to "Offset Edges", otherwise "Straight Edges".
- **"STONES":** these settings govern the range of the size of the stones. "Depth" will also set the thickness of the wall. The stones may exceed these limits at the edges, to make everything fit, or due to the "Row Weight" setting. These settings are for the centerline of the grout, so a larger "Grout W" will reduce the actual size of the stones.
- **"Bevel":** The size of the bevel. If set to zero the stones will have square corners, useful for reducing poly-count.
- **"Grout W":** The width of the grout, pretty straight forward.

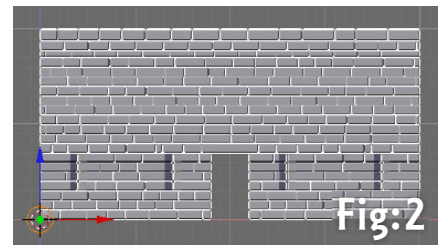
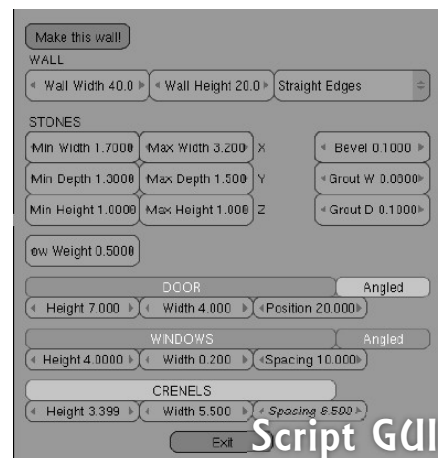


Fig:2

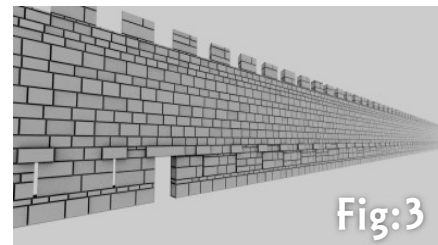


Script GUI



- **"Grout D":** How far the grout is offset from the face of the stones. This references the value "Min Depth" so if there is a large range in depth the grout will be quite far back on the thicker stones.
- **"Row Weight":** This setting governs how much the height of the row affects the width of the stones in that row. Positive values are normal (taller stones, wider stones), but negative values are valid as well (taller stones, narrower stones). I usually set it between 0.0 and 1, depending on the application.
- **"DOOR" and "WINDOW":** These settings behave very much the same. Turn on and off a door or windows by pushing the big button labeled "DOOR" or "WINDOWS" respectively. Door "Position" is from the origin to the centre of the door. Height and Width behave as you would expect. Window spacing will be centred between the edges, and the edges of the door (if one is present). The top of the windows will be at the door "Height" setting, even if doors are turned off. This means that door and window lintels have the same height (generally true in architecture). The "Angled" button toggles an angle along the sides of the door/window, such as would be present in a castle arrow slit. Generally, doors do not have angles like this, but I left it as an option, just in case you want it.
- **"CRENELS":** The crowning part of the wall, crenels are often seen on castle walls. Height, width and spacing work the reverse of windows, setting the size of the stone portions, not the openings. Crenels are always angled, though I'll add an option to control this in the future.
- **"Exit":** This button fires the masons and lets you do the work yourself. Very useful if you are done creating stone structures just at present.

The script is rather fast, so you can make very large walls if you want. I think the wall in figure 3 took thirty seconds to generate.



## A Few Techniques

Here are a few tricks I've come up with. I suspect ingenious users will come up with many more.

- **Paths:** Make a narrow, tall, wall with no windows, doors, or crenels and set it to follow a curve on the Z axis. It's a path, street, or cobblestone highway! See figure 4
- **Support Beams:** Crenels can also be made to hold support beams for multi-story castle towers, siege defense structures, etc. See figure 5.
- **Multi wall:** You can build a wall in multiple segments. This allows several different styles of windows and doors stacked or strung together. The example for the Support Beams was actually created by stacking two walls on top of each other.  
**Sub Surf:** The script automatically sets all edge crease values to 0.5 which results in a pretty good Subsurf results.
- **Texture bake:** Render the Z value of the wall on an isometric view and save it to a texture. Now you can use this as a bump channel or texture mask for a low-poly wall. Using "Offset Edges" and correct clipping allows seamless tiling!
- **Multiple Curve Deformers:** Often walls are tapered or curved. By applying a Z-axis curve deformer above the X-axis one, many wall effects can be achieved. This can also be used to make more realistic streets (see the curved profile of the path example), or cartoonish towers.

I hope this script allows many of you to realize your dreams of realistic castles, brick walls, roads, textures, towers, and anything else you can do with it. I'd be thrilled if you'd drop me a line and let me know what you're using it for. Let me know if you run across any bugs, and I'll try to fix them right away.

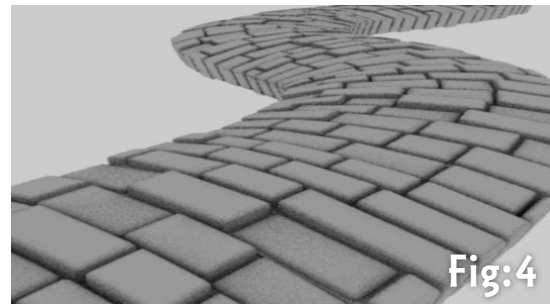


Fig:4

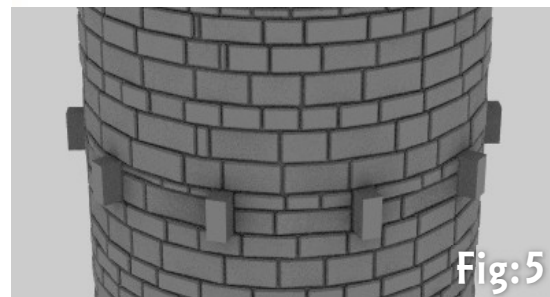


Fig:5

## Writing the Script

If you are still reading this, you may be interested in how the script was programed. The code is decently well documented, so feel free to jump right in and fiddle around with it. Here's a little background on how I wrote it as well as a few lessons I learned. Some of this is just good programming practice, some is specific to Blender.

Before I could program anything I had to learn Python. In college I took a computer programming course, and I've done a bit of programming in my spare time, so there was some background to build on. Essentially, all programming languages are expressions of a logical progression, and Python is no different. I worked through the built in "Getting Started" tutorial in Python, wrote a few small applications to play with features, and got comfortable with the syntax. At this point I started to ease into Blender python extensions. My first Blender script took a mesh object and created every possible polygon with the existing vertices. Not too useful, but rather fun. I also discovered that you can crash Blender by trying to make a mesh object with too many polygons.

At this stage I was also examining a lot of the scripts included in Blender. Here are the first two things I learned about programming Python extensions:

**Lesson 1:** Do your homework. This is true in every aspect of life, and applies here, too. All the basic python tutorials I did were invaluable in seeing how to structure an efficient script. Looking in the Blenderartists.org forums helped a lot too. If you put the effort into discovering the right way to do things, it will make your

job a whole lot easier later on. Learning Python before learning Blender Python really helped to keep things straight.

**Lesson 2:** Document your code. This is one of the big stress points in computer science, and it is so helpful! If someone else wants to use parts of your script, they shouldn't have to reconstruct what every line of code does. If you put the script down for a month or two, you won't remember what anything does, trust me. I know it's a drag. I know it feels pointless. Please, oh, please, document your code. When looking for examples, documentation helped me out immensely.

Once I had done a few mini-scripts in Blender I started on the wall maker. As the script progressed from making plain squares of wall to complex walls with doors, windows, and edge features, I noticed a few other things:

**Lesson 3:** Programming Python extensions is not as hard as it sounds. The extension framework in Blender includes some very advanced tools which makes programming extensions rather elementary. If you have an idea for a script, and have any confidence at all in your programming skills, give it a shot!

**Lesson 3 Corollary:** But it's harder than you think. Be prepared for frustrations. Even though it's nice, the python-blender integration is far from perfect. The logical way of doing things is often (but discouragingly, not always) how it will work. Be prepared to persevere through the tough spots. While working on the script I had three other windows open.

The Python Scripting reference (accessible from inside Blender in the Help menu), the Python package help menu, and the Blenderartist.org scripting forum. These three, with lots of persistence, should guide you through your seasons of scripting woes.

As the Script progressed from simple to complex, I found myself discovering that my original architecture was insufficient or highly inefficient for what I was trying to do. I think I re-wrote every function at least once from the ground up. Initially I was using the "Bevel Center" script to do the beveling, but I had to re-code parts of it to make it work from an outside call (which would require downloading an altered version of Bevel Center along with Auto Masonry) and it was very slow. In the end I bit the bullet and re-coded my basic block function to include beveling. At the beginning of the project I had a free form approach, which I abandoned after working a few days on it. Here's the Fourth lesson:

**Lesson 4:** Be willing to re-think your approach. Even if it means starting over. Every so often step back and say to yourself, "Does it make sense to do it this way, or is there a better way?" If you want a good script, always opt for the better way. Sometimes it won't do quite what you had in mind, but that's better than it not doing anything at all.

I still have lots of features in mind for future development. I'm also thinking that I will have to re-write a lot of the structure when I get back to the project (in line with lesson 4). However, once the script was mostly finished, it was time for:

**The GUI!** Writing the GUI can be the most fun or most heartbreaking part of the process, depending on how

well you have managed your variables. If you have nicely written functions and variable management, the GUI should be pretty straightforward. Writing a GUI in Blender is pretty easy, and really finishes off the script. On the other hand, don't feel compelled to include every feature in the GUI if it isn't well supported, or reliable. There are several features in Auto Masonry which are almost functional in the code, but not in the GUI. If your code is well documented then code-heads will be able to use these features anyhow, and not including them in the GUI helps reduce headaches for entry-level users.

## Future Improvements

I'd like to improve this script in several directions. Unfortunately, my time is absorbed with other projects right now. Here are a few of the things I'd like to add. If you have ideas for other features please let me know:

**Arches:** Right now windows and doors are flat topped. I'd like to add both round and pointed arches. This will require significant re-coding of the basic architecture of the script, but I think it will be worth it.

**Flaws:** Not all stones are perfect. Some stones should have missing corners, or even cracks. An early version (while using Bevel Center) was able to knock off corners, but I haven't added that feature since converting to an internal beveling scheme. Adding cracks would be even more difficult, but I'm sure it is possible.

**Grout:** Right now the grout is simply a second set of blocks behind the stones. On curved surfaces this creates problems with the grout, which Subsurfing helps, but doesn't fix.

A true stone grouting algorithm would also require a major overhaul of the code, but would also create a much better finished product, and would allow beveled grout.

**More robust window and door placement:** Right now the GUI for the window and door placement is rather limited. The script itself can accept any configuration of window and door placement, but I couldn't figure out how to simply implement the user interface. If there is a good way of doing this, will someone let me know?

**Persistent settings:** the ability to save settings to a file would make it easier to tweak settings. It would also allow the creation of multiple walls with "offset edges" that will mesh with one-another, or angled corners with interlocking edges. Unfortunately, this would also require a similar code overhaul.

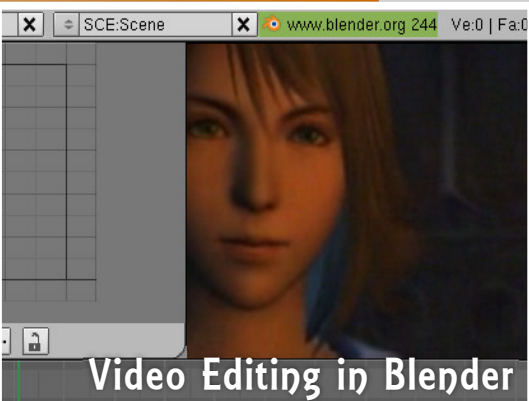
So basically, when I am feeling up to another few weeks of programming, I'll rewrite the script to be better than ever. When I do, I will try to include any feature requests I receive in the meantime. Alternatively, if someone else is feeling bold, I encourage you to make changes to the script yourself. I'd be happy to learn of all the improvements that can be made, and I'm sure there are many.

I hope this script allows many of you to realize your dreams of realistic castles, brick walls, roads, textures, towers, and anything else you can do with it. I'd be thrilled if you'd drop me a line and let me know what you're using it for. Let me know if you run across any bugs, and I'll try to fix them right away. •

***Happy Blending!***

by Paul Spooner





Abhishek  
Gupta  
India



I am currently pursuing C.A. (chartered accountant) which does take my most of the time. But thanks to Blender it has given me the power to pursue my dreams.

[abhix@yahoo.com](mailto:abhix@yahoo.com)

## Introduction

This may come as a surprise to many new Blender-heads, but yes, Blender has a video editor. This may be the most ignored and least developed feature of Blender. But as I recently found out, it can do its job quite neatly when combined with the node based compositor; it packs a mean punch.

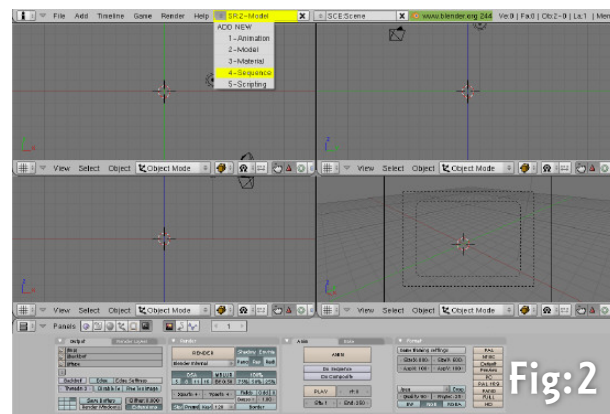
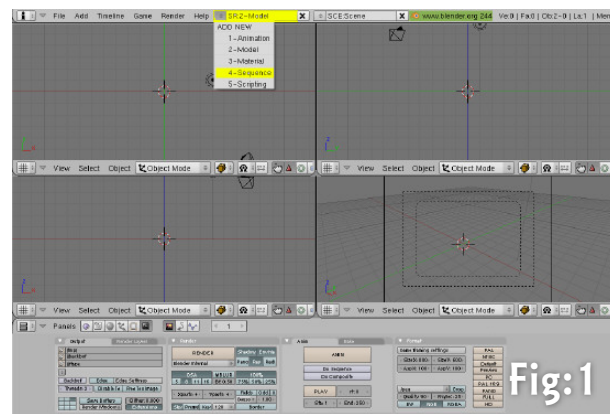
**Requirements:** This tutorial is aimed at beginners, so I hope even those who have never done video editing or have NEVER used Blender should be able to follow it.

## PART ONE: THE INTERFACE

When you actually kick-start Blender, its default interface is designed for modeling. Therefore you have to change the interface first in order to start your editing. Thankfully, there is a preset interface for video editing in Blender. On the upper-most header, there is a drop down menu where "SR:2-Model" is written which represents "Screen 2 for Modeling". Drop the menu and select "4-Sequence" (fig. 1).

Ta-da! Welcome to the Sequence Editor. Let's check out the interface. On the top-left side is the IPO Curve Editor. This allows us to modify lots of effects, like if you want to slow down the time at a certain point, increase or reduce certain effects, etc. For this tutorial, we will just leave it as it is (but you

can play with it later on). At the top-right is your viewer where you can preview your video. At the bottom is your actual video editor. Below is your timeline. Of course you can change the interface as you like (fig. 2).



In the editor panel press "Add" and select "Movie". If you want to add audio and video together then press "Movie + Audio (HD)" (fig. 3). Browse to and select the video file you want to edit. As this issue is about fantasy, I've selected a final fantasy video. He he! Once you have opened the file a blue colored strip will appear. As you move it left or right with your mouse, you will notice some numbers changing at the left-most side; these are frame numbers. Drag the strip toward the left until you reach the zero frame and left-click.



Now if you press "Alt+A" on your keyboard you will be able to play back and view the movie in your viewer. As the movie plays you will see a green line moving forward and your cursor changes to show which frame is playing. Press "Esc" to stop. If you press your left or right arrow keys you will scroll through the frames. You can jump directly to the desired frame by clicking in the sequence editor. Check out the last frame number and note it. Then, move ahead some frames and hit "F12" to see your final output. BOOM! What did you see? Nothing? That's because we have not yet told Blender's renderer that we wanted to output a sequence. Press "F10" to go to the Scene buttons in the bottom-most button window and press "Do Sequence" in the Anim tab. Now change the end frame number to the same as the last frame number of the video which you want to edit. Now press "F12" and you should be able to see your video in your render output.



That's right... time to do the dirty work. Here is what we want to do:

1. Delete some scenes.
2. Add transitions.
3. Increase the video resolution.
4. Add a diffuse glow effect.

1. **Cut it:** Select the strip by right-clicking on it. It should turn dark blue. Remember, whenever you want to edit you will have to select the strip. Now move ahead either using the arrow key or clicking on the editor. When you reach the desired frame press "K" on the keyboard. Now you should have two strips. Move further ahead and press "K" again. Now you should have three strips. Select the middle strip and press "Delete" to delete it. This way you can delete the scenes you don't want to keep (fig. 4).
2. **Smooth it:** Now if you play back the movie there will be gaps between the strips. In order to remove the gaps and smooth the flow from one strip to another, we will add transitions between them. However, I first want to convert all the strips into a single Meta Strip. What's a Meta Strip? A Meta Strip can act like a normal video strip but it can contain many video strips inside it with different effects applied to each of them. Why use a Meta Strip? Well I could have done this edit even without the Meta Strip applied but the Meta Strip is going to help me later on. How? Well, wait and see. Now select all the strips by holding down the "Shift" key and right-clicking on each strip. Now press "M" followed by "Enter". You just created a Meta Strip (fig. 5).

You can toggle inside or outside of the Meta Strip by pressing the "Tab" key. Now let's go inside to add the transitions. Select the first two video strips. Press "Add"; go to "Effect", select "Cross", and then left-click to confirm. You should now have a red strip between the two video strips. This is the cross-fade transition strip. Select the second strip and grab it using the "G" key and move it up a bit and drag towards the left side until it overlaps a little bit of the first strip (see the figure). Now you should have a smooth transition between the two scenes (fig. 6). Repeat the same procedure with the other strips. You can also choose other transitions like "Wipe".

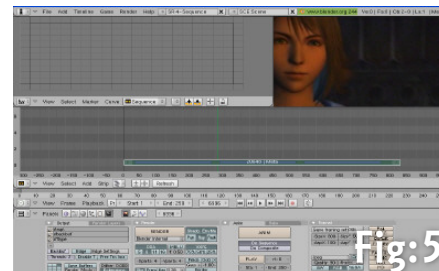


Fig:5

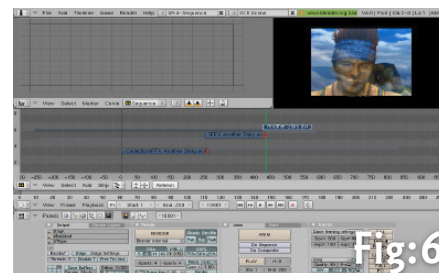


Fig:6

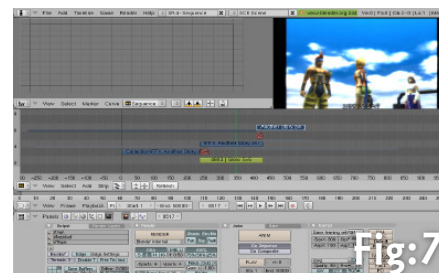


Fig:7

3. **The effect:** I wanted to add a diffuse glow effect in a particular scene which I have already cut. So select the middle strip, press "Add", go to "Effect" and select "Glow". Move the strip just above the video strip and left-click to confirm. You can use the "G" key to move it. Now if you play the frames of that video strip you should have a nice glow effect at the highlight areas. If want to change some parameters regarding the glow effect, select the glow strip and press "N" (fig. 7). Change it as you like and toggle out of the Meta Strip by pressing the "Tab" key.
4. **The secret of Meta Strip:** Here is the reason why I wanted to use a Meta Strip in the first place. The video I have is of VCD quality and I wanted to render it out in DVD quality. To compensate for the loss of quality, I wanted to add a box blur filter. If had not used a Meta Strip, I would have to apply box filter to each video strip. This could have turned out to be a very time consuming process if I had to edit many individual strips. Now that I have a single Meta Strip which contains all the edited video strips, all I have to do is apply the effect to one Meta Strip only. Therefore, select the Meta Strip and press "Add", "Effects", "Plugin" (you may need to download this filter which is easily available; Windows users will need to download the ".dll" files while Linux users will download the ".so" files). Now browse to the directory where you saved your plug-in and select box blur filter. You can also change the parameters of this filter by pressing the "N" key. I reduced the size slider from 5 (the default) to 1.
5. **Render it:** In the buttons window at the bottom-most screen, press "F10" to switch to the Scene buttons. In the Format panel, select the desired format you want to render your video out to. I selected AVI Codec and chose XviD codec. Linux users can use "FFMpeg" to render in a compressed format. In the "Size X" and "Size Y" buttons, click and enter the desired resolution you want to render to (note: you must enter a resolution of at least the same, or greater than, the original video's resolution). Press the "ANIM" button to render your video file.

There you have it. Your final video output done entirely within Blender, and yet we have not even touched the advanced features. Blender is capable of much more than this. •

Note: I have not yet managed to get audio from Blender's Sequencer Editor. I have used Jahshaka for that purpose. Jahshaka is a video editing and effects system. It's also an open source product and is pretty impressive. The latest release is version 2.0 RC4, check it out as well!

# MAKING OF: 'Andy Tear' liquid tubes

30



Andy Tear Liquid tubes

## Introduction

Hi there. Many people have asked me about how I created the tube effect on my "Andy tear" illustration. As I always work step-by-step, I will talk about the different steps I used. First I always finish modeling before something else. I think it's not good to mix different steps like modeling and material, or lighting and material etc. I think it's important to cut the work into different parts. For an illustration, here is my work flow: modeling, bone setup, skinning, material

colors (only color), environment and lighting (render options, like AO, AA etc.), materials (with nodes if necessary), rendering effect (I always make this part with different layers and importing them into Photoshop, but you can translate them into a Blender compositing node setup). So here are the steps for the tubes :

## Modeling

First, in Top view **[NumPad 7]**, create a Bézier circle **[Space] / Add / Curve / Bezier Circle** and name it "CurveBevC1" (for future use). Always remember to name your objects in Blender. It makes it easier to find them in the Outliner window. In Edit mode **[Tab]** duplicate the circle in place **[Shift + D] / [Ctrl]**, and scale **[S]** it into the first one. Make it again to form your circle curve like fig1:

Now, in Left view, **[NumPad 3]**, create a second curve like an "S". This curve will be the path of

"CurveBevC1". Name it "CurvePath1". Don't forget to enable the 3D property of the curve using the "3D" button on the "Curve and Surface" panel. It allows the curve to have 3D coordinates for it's control points. You will obtain something like in fig2:

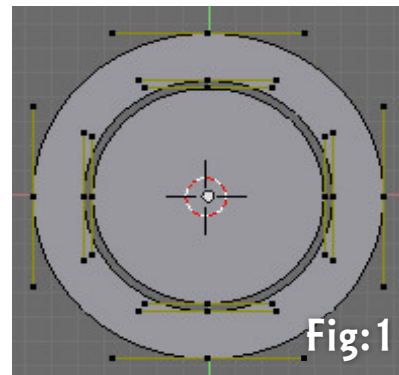


Fig:1

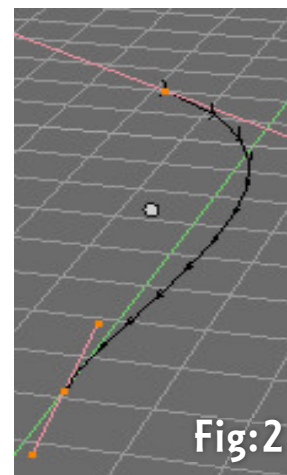


Fig:2

by Cyan

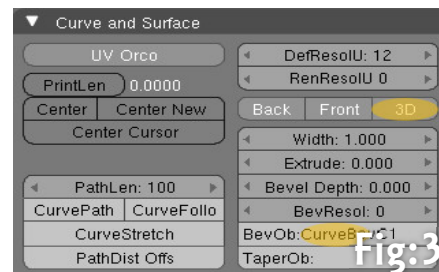


# MAKING OF: 'Andy Tear' liquid tubes

31

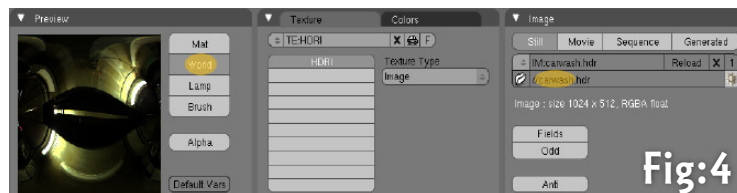
Add the "CurveBev1" name into "BevOb" text box (see Fig 3).

The question is : Why have I made a tube with another tube not joined to the first? Because we will add 2 transparent materials, one for the tube and one for the liquid. Refraction can cause artifacts with two joined meshes. Now to finish, convert it into a mesh [**Alt + C**]. You can also close the liquid (extrude an edge-loop and merge the vertices [**Alt + M**]) and tube hole (select an edge-loop and make a face [**F**]).



## Lighting & Environment

We will create reflect/refract materials, so to create a reflecting environment I like to use an HDRI map mapped onto the scene environment. To do this, go to the **Texture buttons [F6]**, and create a new World image texture. Select "carwash.hdri" (or another) like in fig 4.



After that, go to the **Shading / World buttons [F8]** and set to "Sphere" mapping on the "Texture and Input" panel. On the "Map To" panel press "Hori" and set the blending mode to "Add". Set the World color dark, and with the "Col" slider you can tweak the HDRI map intensity. See fig 5.



You may tweak the lighting as you wish. I always work with a lot of quad attenuated lights to create a smoother atmosphere. Add some indirect orange lights, and blue lights for the back lighting. For main lights choose a high-level white lamp. In my lighting work I always work each sort of light step-by-step. Example: if I am working on the back light of a 3-point lighting set-up, I isolate the back lights and inactivate the direct ones. I make a lot of tweaks on the back lights, do a lot of renders, and when I'm satisfied I work around the indirect orange lights, and inactivate the others. I think it the best way to have the most control over lightning.

by Cyanb

## Materials

Before creating materials, I've assigned 3 material IDs to the mesh. The outside of the tube, the inside, and the liquid. See fig 6.

So select the different parts of the tube and assign the 3 new sub-material IDs. Outside is a transparent material with thin bump and a refract coefficient greater than 1. See fig 7. And the thin bump setup as in Fig 8.

Inside, I've duplicated the first material and removed the bump. Try a different refraction factor, and see what happens.

For the liquid I've create one purple material with "Ray Transp", and a high amount of "Filt" (transmissivity). It permits you to filter light by the material color, and make the liquid appear purple. For all these ray-traced materials, don't forget to increase the ray level up to 6, because one ray must pass through 6 mesh surfaces before reaching the camera. See fig 9.

I've played with the "Ramp" color ("Input" set to "Normal" and "Method" to "Add") to simulate the lighting that passes through the material. Translucency and color emission are really important too, to tweak this effect. See fig 10.

## Post Processing

For post processing, I will not go into all the details because the glow method I've used does not involve Blender. I've just exported color layers to select the purple colors rendered. After making my selection, I've duplicated the image selection in another layer in lighten mode and blurred it. I've already simulated a little reflection and refraction in objects. To achieve the final effect, I've duplicated the layer several times.

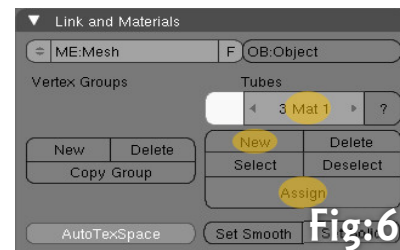


Fig:6

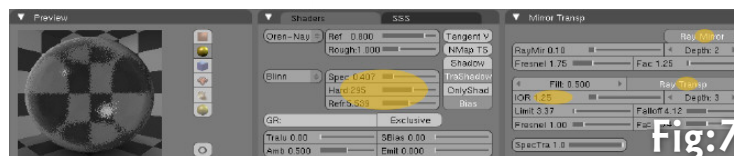


Fig:7

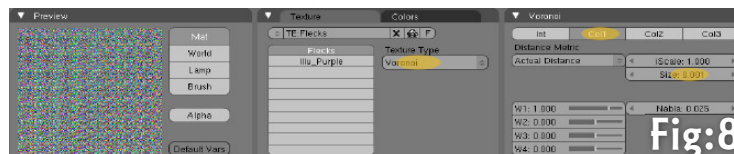


Fig:8

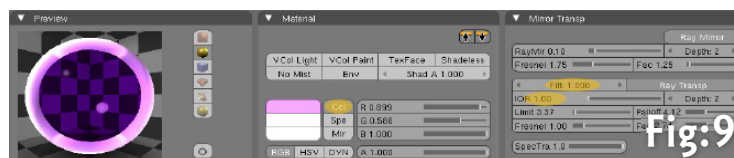


Fig:9

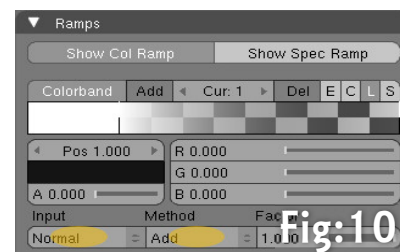


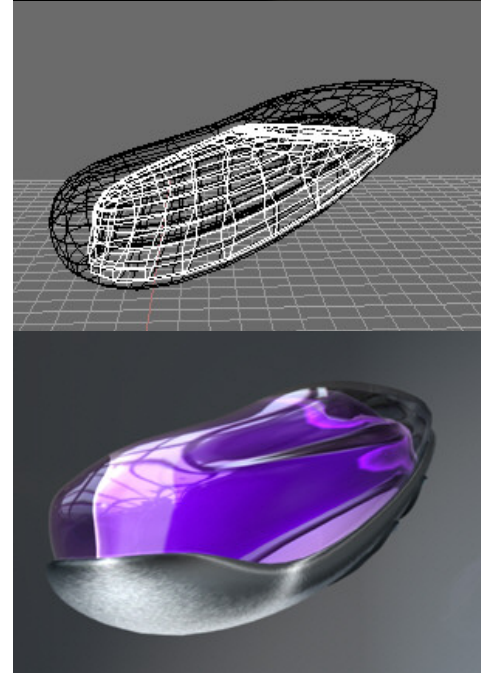
Fig:10

# MAKING OF: 'Andy Tear' liquid tubes

33

But you can also use the "Video Sequence Editor" or "Composite Node Editor" to add glow. So this is the end result as seen in fig11.

It's the same technique I've used for the head in my illustration, except for the bump on the outside container. •



by Cyanb



Glass Girl

**Kamil Latocha**  
Poland

(aka migero)  
I started using Blender since version 2.25 was born, while it evolved I also learned 3dsmax and some Maya, but Blender beats them all.

## Introduction

At first the project name was 'Glass Girl' and I wanted it to be fast and simple. As time went by I thought, "It's starting to look good, why just leave it like this if I can model higher polygon meshes and test out my baking skills?"

Because I had already made a body in pose (not in a standard T pose), it was hard to make her very symmetric and keep good proportions of body/head parts. Before you start a

new project it's helpful to think about what you actually will do to avoid problems later.

Glass Girl's mesh is really crappy here but I'm still learning to model like a pro. ;) There's not much to tell about modeling her, as she started from a box and ended up as a ~1000 polygon mesh, which was good for sculpting. I didn't use the multires function (as I didn't have UV there yet) but just used grab, inflate, and pinch to fine tune my modeling.

## Wings

Wings were a real challenge to me, as I didn't know how to start modeling them. First, I made a base wing and used dupliverts with feathers. Next, I made a base wing and just extruded edges to mimic feathers. Unfortunately, both methods gave poor results.

I found a reference from Wikipedia about bird wings, and I started modeling to make them manifold, or sufficiently varied, and give the feathers some thickness. I knew it would be a pain to UV map it, but in the end I used plain unwrap and helped by adding seam edges around feathers and at their joining with the wing bone. To model so many feathers, extrude the faces individually (in edit mode, select all faces and use E key - individual faces option) (see fig 1) Start at the top of the mesh and select enough faces to extrude 2-3 rows of feathers. Adding some random size, length, and rotation adjustments is a good idea.

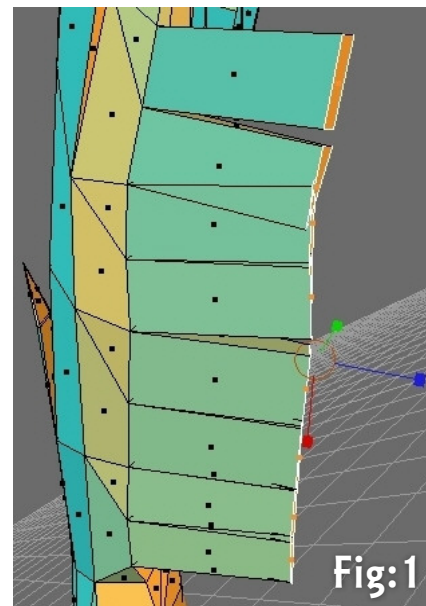


Fig:1

## Detailing feathers

The next step is to detail the feathers. Select all their side edges and subdivide them 3 or more times, depending on their length, until they start looking like in fig 2. If you want extra details on some of the feathers you can edit them this way, see fig 3.

## UV for wings

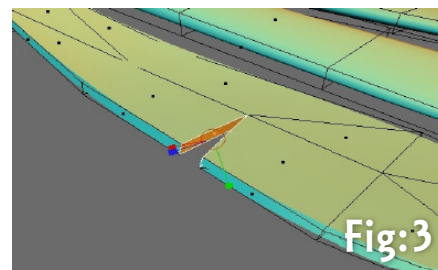
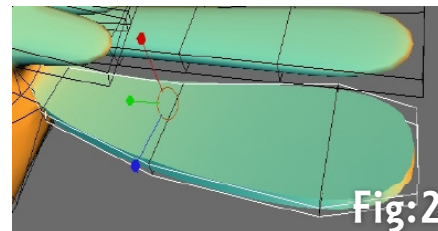
Remember that UV maps for the wings are only needed if we want to bake them. Before you do this on such a complex model you need to be 1000% sure that the mesh is correct. I made 3 UV maps for my wings, and the last one seems to be ok, so don't get angry if your first doesn't look good.

First, (in edit mode) hold the shift key and start selecting side, tip and base edges of feathers. When you are done selecting, hit alt+E and select "make seams". Now create a new UV map for it. The more the complex the wing is, the bigger the size of map should be. ;) In UV face select mode, selecting all faces, pressing U, and picking unwrap will do the trick.

If you are not happy with the results (if some islands of feathers are too small, they are very chaotic and use less than 50% of the map space) then I suggest you use pack charts in UV/Image editor. If it still is not good, you can always rotate/scale/move UV islands by hand. Alternatively, if you have the python script for smart unwrap, use it (script found in blender 2.44 but it doesn't pay attention to marked seam edges and creates it's own) and play with the settings. Incidentally, in such a complex mesh I do not suggest using lightmap Uvpack, but if you change the map size to 4096X4096 then I think it will be ok.

## The Cloth

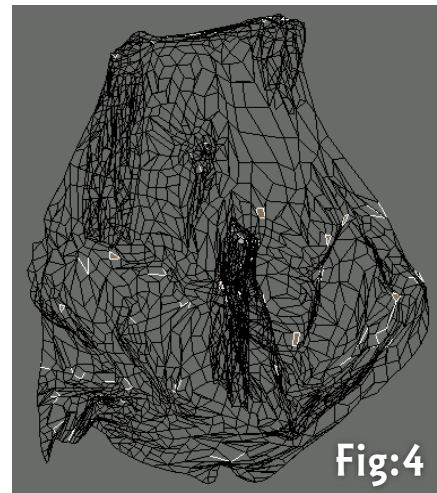
This is where the fun began! I started modeling it from the leg (the one like a stiff rubber, because it had no folds. I started to think of ways to add them. One method was using plain modeling, splitting edges/faces, and slowly adding details.





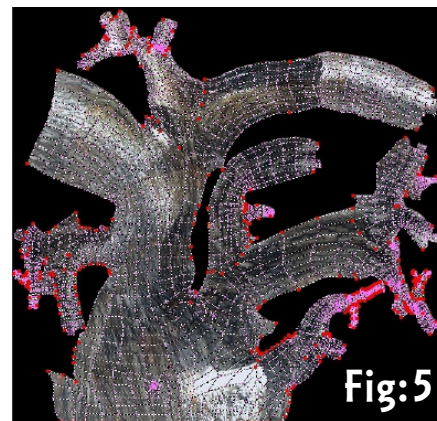
his was fine in the beginning, but in the end it looked very random and I couldn't really see where the folds went - it just looked unreal. Next I tried using a subdivided mesh with sculpted folds, but it became even worse. It's hard to mimic cloth and how it will react this way, where will the folds be, and in what direction they should go.

But I got lucky - while searching over at blenderartists, I ran across a new cloth simulation currently in SVN and some test builds. My only complaint is that it doesn't have self-collision... yet. To create some folds in this cloth so that it doesn't just fall down from that tree, it needed some goals. In edit mode I selected only the main faces and added them to a vertex group. See fig 4. Only 40 frames were needed to make the folds come down and give a really nice effect. After that, I just applied the deformation, and in edit mode adjusted the faces of the cloth that had jumped into the tree's mesh. That's it no more need to sculpt a cloth in Blender ;).



## Tree

While UV mapping the tree was a challenge, modeling was quite fast and easy. UV mapping was difficult because I wanted the tree texture to be seamless from the front (at the back there's only one seam going from the top to bottom and on branches). The real life saver here is live unwrap, transform and pinning vertexes. Just set up the seams correctly, select all of your mesh faces in UV mode (if you have the new test blender, the UV mode tools will be in edit mode), press U, and select Unwrap. Now go to the UV edit window and select the most important vertexes of the tree as shown in here (the red dots). See fig 5.



It's best to use brush select (B key), because sometimes the vertices of the UV aren't joined. In this case, deselecting the pinned vertexes and selecting them again will not move the pinned vertex. When moving the pinned vertices and the UV deforms, be careful - The fewer pins you add, the better. It takes time to learn how it works, but the time is definitely worth it.

The advantage of this method is that you can go back to edit mode and change your edge seams on the mesh, but unwrapping again won't delete your pinned vertexes. That way you don't need to start all over again if you found a missing edge seam.

On the tree I used 2 UV maps. The first UV map is for baking, which can't overlap itself and must be inside the map border. The second map seen here in fig6 is for texture. It doesn't need to be in bounds of the map area as the texture of the tree is seamless.

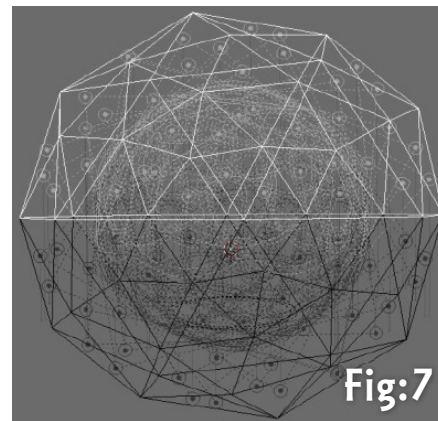
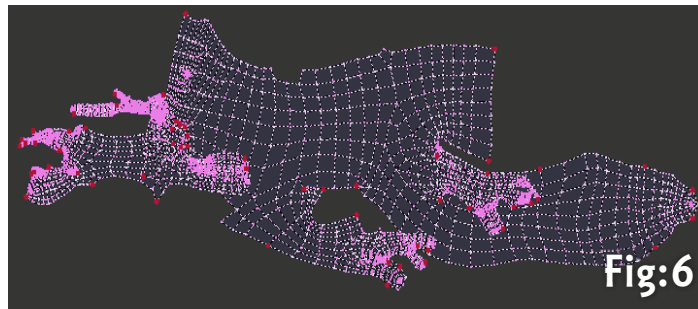
## The Bake

### Part 1: Creating AO bake

To start off, go to left/front orthographic view and Add>Mesh>Icosphere in the center of your scene. Now depending on the subdivision level, you will get more lights (thus, slower baking), so go for a setting of 1 or 2. Higher is not required.

If you want a 2 color AO, then go to edit mode and select the bottom part of the iconosphere ball. Split it (P key), and if you want 1 light just leave it in once piece. In the same place as the sphere, create a spot light, set the light to raytrace and set samples to 1 or 2. Next, parent the light to the sphere. If you split the sphere, you'll need to duplicate the light, leave it in same place, and then parent it to second sphere part.

Setting colors of the lights can be as you want. I used blue and red, but if you use 1 light then white is what you want (as you can always change the color in gimp when compositing). If the lights are parented, select the sphere or their parts and turn on dupliFaces (they seem better than dupliVerts). If the lights on your sphere are facing inside, you're fine. If not, edit the mesh and flip it's normals to make the lights point to the inside. See fig 7.



To bake AO from rays, set-up the material of the object as follows:

- Make the material color white or gray.
- Set the shader type to Lambert and Ref to 1.
- Remove any specular shader (or in lights, set the No Specular option).
- If you want the object to receive transparent shadows, turn on the TraShadow button located in the shaders tab.
- It's also a good idea to set your material with a bump map (if you have a UV map on it) as it looks more realistic.

There are many reasons why I use this method instead of normal AO:

- It can get cleaner results than AO (but is slower if you use sub-div 2 or more of the sphere).
- If using ray shadows on the lamps, you get alpha textures that cast shadows.
- Someone once said AO is for suckers ;) because it uses ray distance, and you can't really set how fast the shadow will blur over that distance.

## Part 2: Sunlight bake

Time for a hard shadow. You don't need to have it, but it adds details to the final composition.

Settings:

- Baked shadows start looking good if the shadow map is 2 times bigger than the AO (so you can scale it down in gimp and get nice aliasing effect on the shadow's edges).

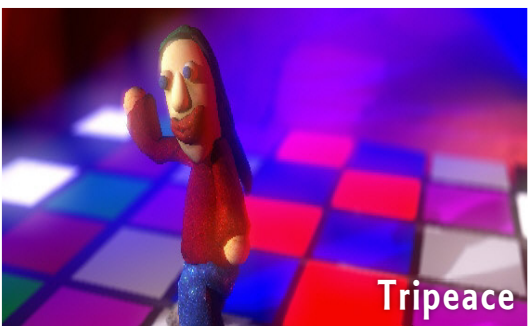
- Adding a bump map here is a good idea.
- Remember to hide the sphere and light we used to bake AO, and create a sun light or a spot light in white color using the raytrace setting. Subsamples is best set at 5.

Things I didn't say:

- After baking is complete, save the bitmap, not the blend file!
- You need to set the light for AO very low. 0.05 will be fine. Also, set the distance way up (in new test blender there is a constant fall off of light intensity which will never fall to black).
- How to compose the maps in gimp (just using layers and mixing AO+Shadow+Texture). It all comes down to your skills as there's really no recipe on what types of mixing to use. •

# MAKING OF: 'Tripeace' animated

39



## Introduction

Initially, we just wanted to create a short film of 2 minutes (with neither a scenario nor history, just for fun) to give happy feelings and pleasure through images, bright colors, and simple style. We took such great pleasure in creating these two minutes, that once it was finished it ended up being 11 minutes! The original idea was to make a journey through time and space, and music

about evolution. "Tripeace:" a conglomerate of "Trip" and "Peace" :)

In terms of design, we wanted a simple style, and we thought that modeling clay was ideal for that. We have modeled the characters and have taken their picture from four angles to be able to model them in Blender, and also create textures (cracks, flaws) from the photos of the modeling clay.



What we learned the most in Blender is about Nodes, which can really transform a rendered image. We have used the nodes, "glow," "defocus," and "Bloom FX," which have served us well (maybe we used them a little too much!).



by Denis



# MAKING OF: 'Tripeace' animated

40



Given the extremely long render times (the computer was turned on every night for two weeks for final renders), it is very practical in Blender to first make an animation without lights or textures, to see if the pace and framing is good. Next, put the "master plan" in the edit assembly, do the edit, and only render the frames required.

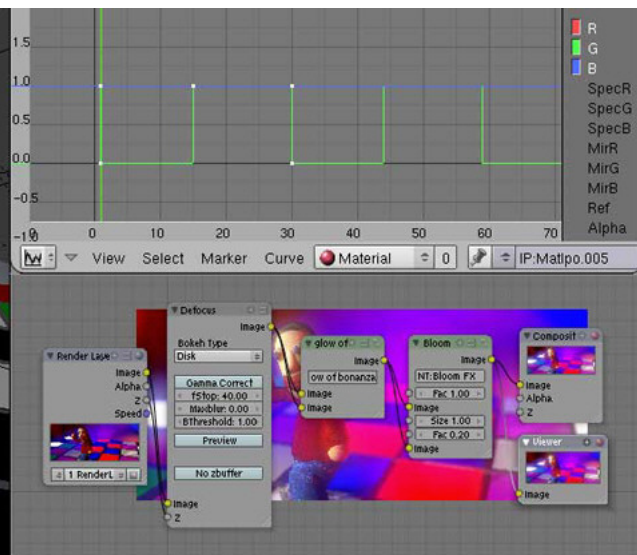
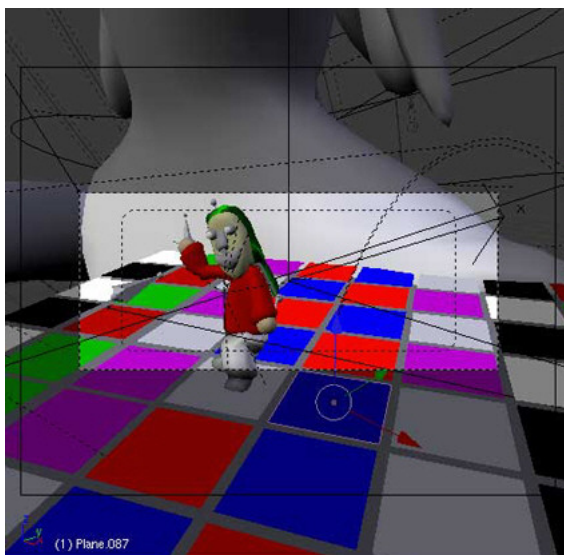
This has been the small making of Tripeace. Thank you again to all the Blender Team for their help and for the very good piece of software that is Blender!

Download Tripeace animation:

On [dailymotion](#) (lower resolution, Flash video)

[High-resolution](#), DivX (embedded)

[DivX download](#) (100MB) •

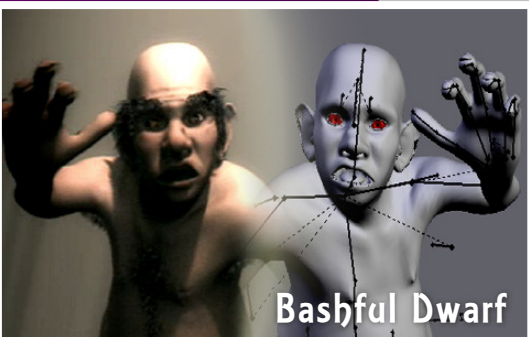


by Denis



# MAKING OF: The dance of the 'Bashful Dwarf'

41



## Introduction

What I like about Fantasy Dwarfs, Gnomes and Leprechauns is that they can be actually funny characters as well as very evil ones. You can never tell what they'll come up with.

Some time ago, when the **Blender Sculpting tools** came into my hands, I decided that I wanted to build a really nice humanoid character. I started thinking about the different

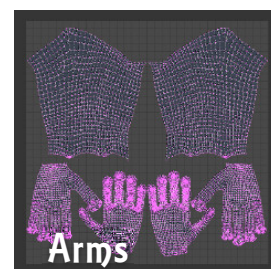
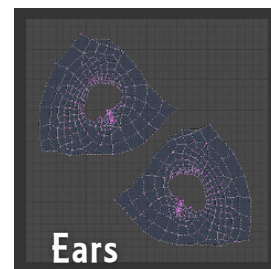
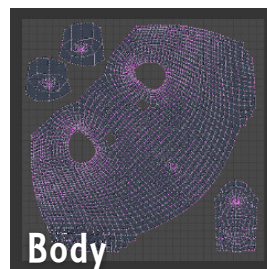
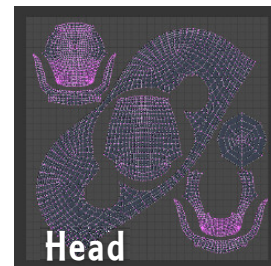
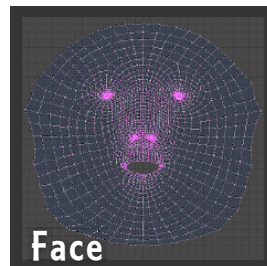
options, and finally I realized that the best I could do was to make a character whose main characteristics resided in the proportions of its body.

I didn't want to make a standard "Muscle Man", so then, a drawing that I had done when I was a kid came to my mind, and I knew that I had to model a shrinky, old fatty dwarf!!

I took a standard humanoid model I had made a few years ago, and I started sculpting it. Some topology refinement had to be done too. Finally, after a couple of days, my Dwarf sculpture was finished.

The next step was to **UV Unwrap** the model. That was quite easy with Blender tools (Unwrap and Proportional Editing). Then, I carefully started arranging the **UV Layout**. As I knew I was going to use bitmaps for the main diffuse textures, **I planned a UV layout that would allow me to have really fine detail in the face without compromising the details of the rest of the body.**

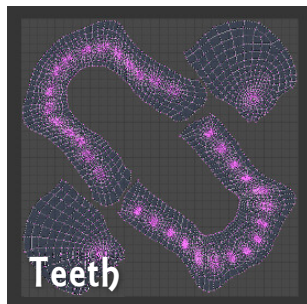
As you may know, it is almost impossible to work with a 10K texture, so I strategically divided the UV layout in different sections. This way I could texture the body using separated smaller bitmaps. I finished up having seven different layouts.



by Juan Pablo Bouza

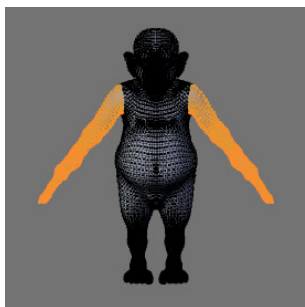
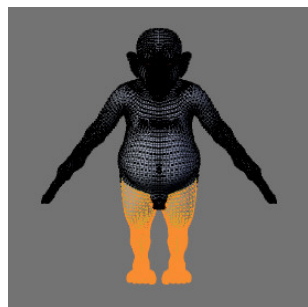
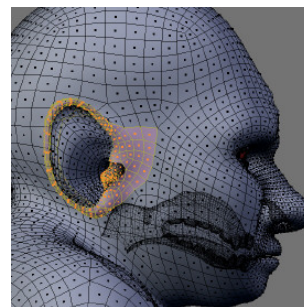
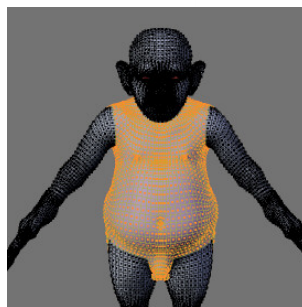
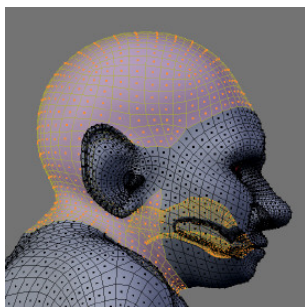
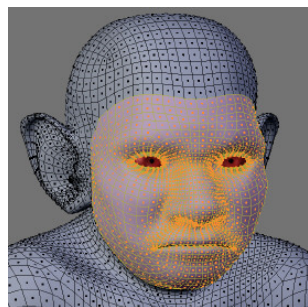
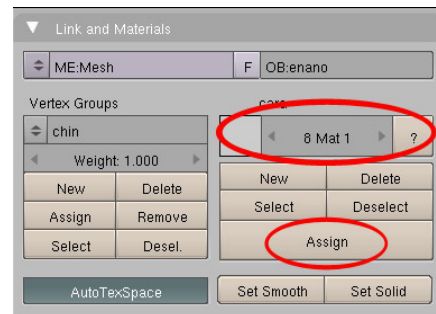
# MAKING OF: The dance of the 'Bashful Dwarf'

42



It is important to say that each layout corresponds to a different **Material ID number**. This way you can assign different bitmap textures to each different part of the body. In order to do so, what you have to do in this case is to make all the different seven materials. Once you're done, you go to **EDIT Mode**, link and materials panel and select Material number 1. Then select all the faces that correspond to that specific UV layout, and hit the **Assign** button. Repeat with all the other

different materials.



by Juan Pablo Bouza

I started working with 4096x4096 images; actually, I did all of the texturing with images of that size. However, when I started making the first rendering tests I realized that having seven 4K images for the body diffuse textures, plus the Subsurf modifier, plus the Displacement textures (which I will refer to later in this article), plus the Particle Hair and the Bump maps, was a little too much for my 2048 Mb of RAM computer (especially when I tried to Render 4K images).

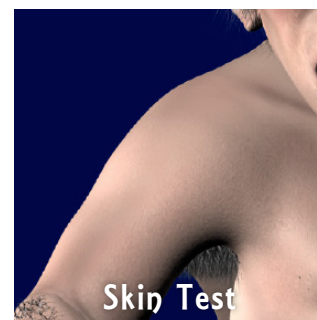
So, I downsized the textures to 2048x2048 and the memory usage lowered drastically. Fortunately enough, the smaller size of the bitmaps did not have an impact on the texture details, not even in a close up look of the character. Well, after carefully painting the model, I finished up the main color textures.



## It was time for the Bump maps!

At first, I intended to paint the bump maps myself, but there were two things that stopped me from doing so. The first reason was the RAM usage, and the second one was my lack of patience for texture painting!

So I finally decided to save some RAM and went for a **mix between procedural mapping and bitmap textures**. Basically the pores of the character and the skin textures were achieved by using two different procedural materials, **Musgrave** (for the pores) and **Musgrave Plugin** (for the roughness of the skin). These two textures (Right bar) gave the main "skin feel" to the character's surface.



# MAKING OF: The dance of the 'Bashful Dwarf'

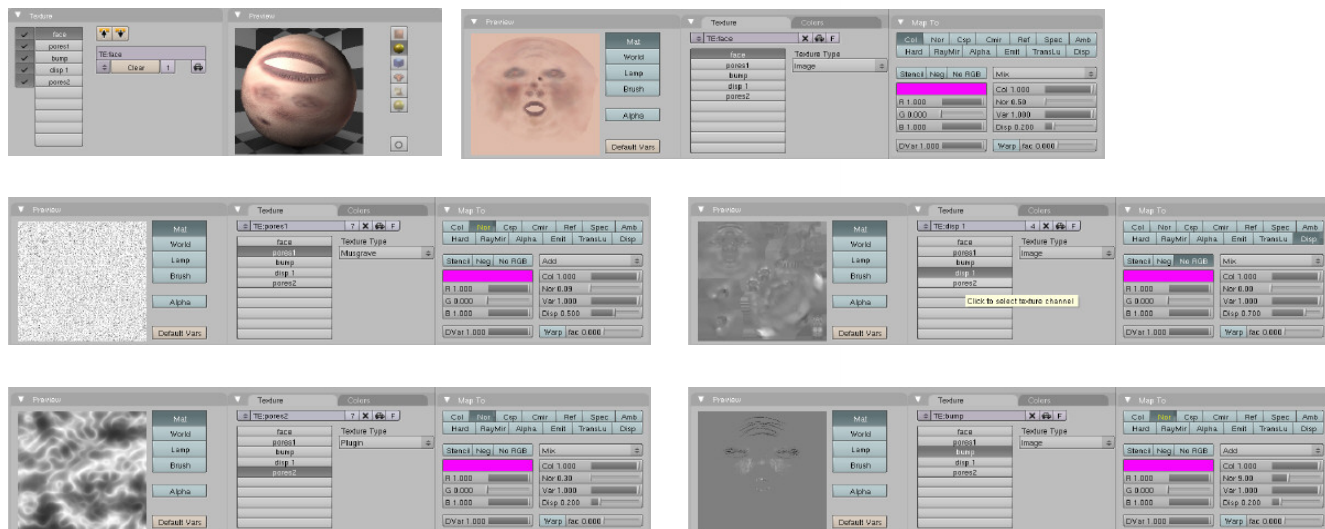
44

But there were some details that I could not possibly achieve with simple procedural mapping; **the Wrinkles**.

For the wrinkles details, I used a very simple 50% gray bitmap, and I painted the wrinkle lines over it. The darker the lines, the deeper the wrinkles. Then I added this map to the material as a Bump map.



This is the Final layout for the **Face** material:



by Juan Pablo Bouza

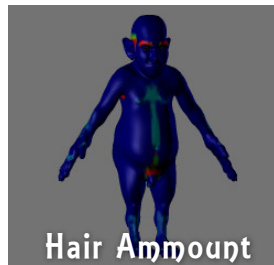
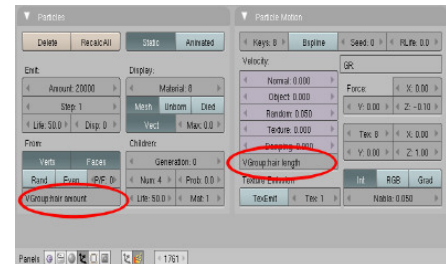


# MAKING OF: The dance of the 'Bashful Dwarf'

45

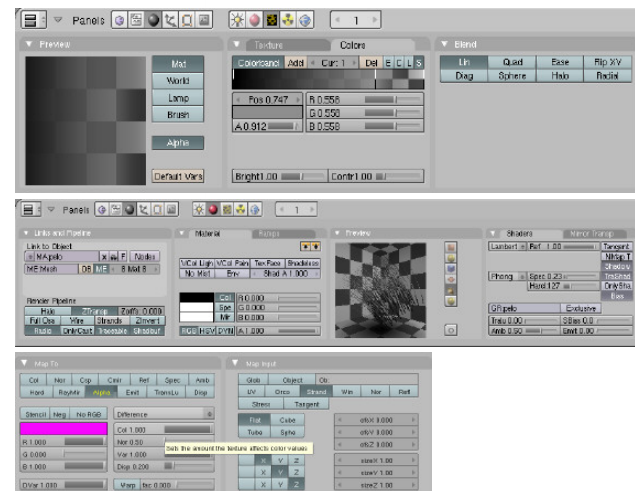
Finally, for the **hair**(see image on right), I made two new **Vertex Groups**: one for the hair density, and another for the hair length. I enabled **Particles** on the model and set up this layout (which I took from the **Blender Wiki tutorials**. You should read all of it before you start fooling around with Blender!).

I assigned the two new Vertex Groups to the density and length values of the Particle System, and then I started **Weight painting** these Vertex Groups. The great thing is that the results show up in the viewport in real-



time while you paint. The combing of the hair was fairly simplistic and easy to do. I used lots of **Curve Guides** and a few **Empties** with **Spherical Deflection**, and that was it. (see left)

This is the Material layout (images on right pane) I used for the hair (also from the Wiki tutorials):



by Juan Pablo Bouza



# MAKING OF: The dance of the 'Bashful Dwarf'

46

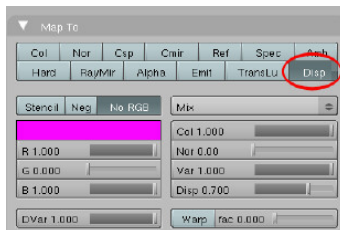
I was so happy with the final result of the character, that I decided to create a rigging system that would enable me to animate the dwarf in any pose I wanted, so I started the **BlenRig** project, but that's another story...

*There is a non-Blender section of this work that I think is worth mentioning:*

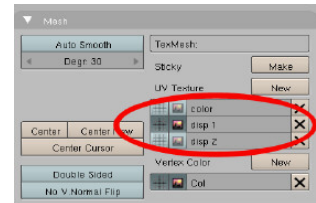
## The Displacement Maps

As you may know, Blender has a superb set of sculpting tools that can be greatly combined with the impressive **MultiRes** feature. The thing is that we are not yet able to properly animate MultiRes meshes, as the MultiRes information gets awkwardly corrupted when you move the mesh out of its rest position. Therefore, deformable objects cannot be animated with MultiRes enabled. The good thing is that this issue is already known in the Blender Projects site. Hopefully, MultiRes Animation will be possible some time soon.

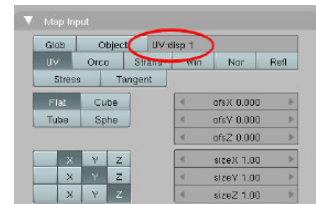
For now, though, the only thing we can do is somehow generate the **Displacement maps** for the model and apply them to the mesh with the **Displacement Modifier**, or with the **Displacement mapping button** in the 'Map To' layout of the Materials rollout ('**Disp**' button).



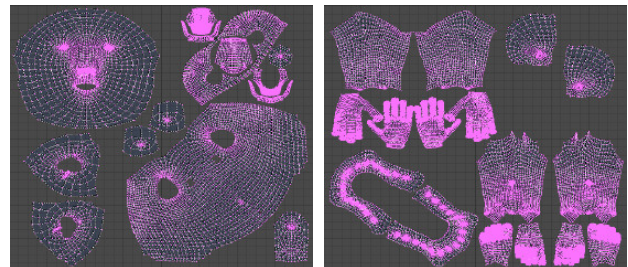
As I said before, I had to save all the RAM I could, and as the Displacement Maps did not need to be so detailed, I created two new **UV layers** just for the displacement maps.



If you work with multiple UV Layers, you must always remember to assign the correct UV Layer name to the map in the "Map input" layout.



As you can see, I scaled down and joined the UV Layouts that I had made for the Diffuse textures. This way, there was no need to have seven displacement maps but just two.

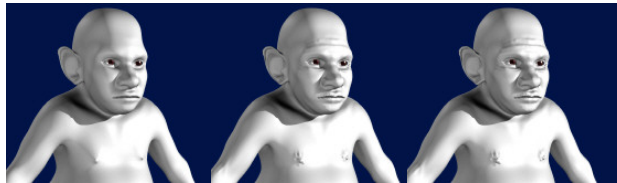


by Juan Pablo Bouza

## But how did I manage to generate these Displacement maps?

Well, there are three (working) free programs I know of that do the displacement calculation job for you. These are: **ORB**, **Xnormal** and **DnormGen** (which has a Blender plugin).

In case you don't know what a **Displacement map** is, it is Grayscale based map that Blender can use to transform the faces of a mesh along its normals (Go to the Wiki and find out more about it). Therefore, you can add very small details to a **subdivided LowRes mesh**. This means that you can animate the lower resolution mesh, and it will render with all the details of the high resolution one.



1 Lowres. 2Multires level3. 3Lowres + SubSurf + Disp Map

After a lot of research, I finally used **ORB** for the Dwarf's displacement maps. It's a little bit old console- based program, but it worked for me.

**ORB needs the mesh to be triangulated, so you'll have to turn the triangulate option on when you export your .OBJ models.**

In order to get the best results you have to play around with the **raydistance** value. In my case I used a value of

2, which did not deliver the best results with the normal maps, but it did for the displacement maps.

As you may know, **ZBrush** and **Mudbox** subdivide the lower res model in order to generate smooth displacement maps. Therefore, the idea is to:

- Model in MultiRes mode and export the HighRes model.
- Afterwards, apply a subsurf modifier to the lower res model before you export it.

That way, ORB will be able to draw a more detailed displacement map when it traces the rays from the High Resolution mesh to the Low Resolution one.

In the case of my Dwarf character, I had to export the Low Res model twice. One time with the **first UV Layout** and another one with the **Second Layout**.

Then, I imported the first .Obj Low Resolution mesh and the High Resolution mesh into ORB and generated the first map. I repeated the procedure with the second Low Resolution Mesh in order to generate the other map.

After the maps are generated, you will probably have to correct the gray values of the seams areas with Gimp, as no matter how little the difference may be, it will show up when you render the model.

As I said before, one of the drawbacks of ORB is that it is based on console commands, another one is that the window must be fully displayed in your desktop, so if you want to render 1024x1024 textures, your monitor

must be set to at least at a 1280x1024 resolution (I would recommend you set it to 1600x1200). Some time ago I gave a little try to the latest version of Xnormal, and I must say that it is working really well now for displacement maps. You could give it a try.

Well, these are the resulting maps. Let's hope Blender can generate Displacement maps from MultiRes meshes some time soon!

## The Dance

Ever since I finished BlenRig, I wanted to make some kind of demonstration of its capabilities, so I imagined this funny scene of this little Dwarf dancing naked in the dark. The idea was to make a simpler project before I went on working on the original short film that the Dwarf was made for, which happens in the woods...

I wanted the movie to have music that had nothing to do with dwarfs, really. At first I thought about some kind of Disco "Bee Gees" style music, but then I came across this nice little piece called "**Jazz Freestyle Instrumental**" ([www.archive.org](http://www.archive.org)) and knew that it was exactly what I needed.

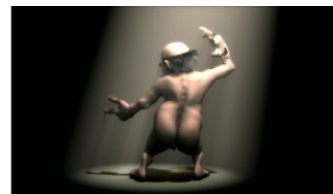
To tell you the truth, the animation was done in quite a hurry. I wanted to make it for the 2007 Siggraph convention and I had only three days left...well, I didn't make it on time, as the final render took about a week. That's the reason why there were some aspects that had to be left behind, but I was quite happy with the final result anyway. Besides, I took it as an exercise more than anything, and I learned a lot from it.

## The Animation Process

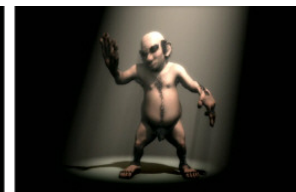
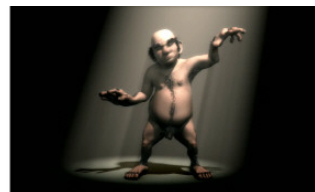
Well, once I had the music set, I started synchronizing the movements. I must confess that many of his dancing moves were inspired by my own dancing experience... sadly enough for him.

So, I started the synchronization process. I would start dancing myself in order to make the choreography sequence; I really hope no one saw me.

The idea was not to make him a great dancer. In fact, I wanted him to be more like clumsy and ridiculous on top of everything. You know, with that "sexy, sexy" thing he is mumbling, and the movement of his pelvis...



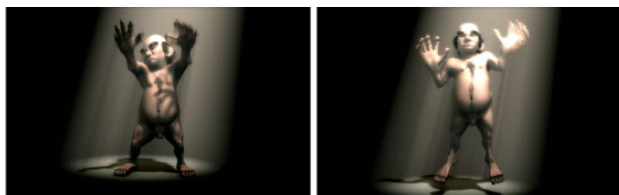
I think that the end result was some kind of mix between great dancing and clumsiness. You have to admit that he has some great coordination in those hands. Try to do it yourself and you'll see that it is almost impossible to do that!



# MAKING OF: The dance of the 'Bashful Dwarf'

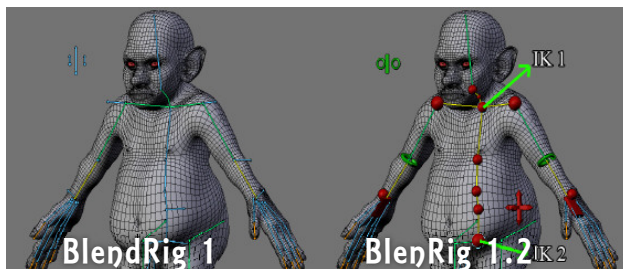
49

So, the first part of the animation had to be really well synchronized with the sound, and I knew that the second part, the jumping one, was more like a freestyle totally out of rhythm dancing.



For the Technical aspects, my main tool was the **NLA Editor**. I did some looping animations, especially for the hip side to side and up and down movements, and I repeated them all over the song.

I had some trouble making the spine move realistically, as I was combining the **IK** of the hips and the legs with the **FK** of the spine. You know, I had to do a lot of compensation movements in order to avoid the spine from moving like crazy when the hip rotated. **This was one of the reasons why I developed the new double IK system for the spine in BlenderRig 1.2.**



In general, all the animation was done using **Inverse Kinematics**. That's why the jumping sequence was not difficult at all to animate. I think that the most challenging phase of the animation was the **turning around moment**.

Full body turning around movements seem to be rather difficult to achieve if you want them to be "**realistic**." The first thing I did was establishing the **upper body's rotation**. Once I did that, I started posing the feet and legs to follow the rotation of the body. It was tough to make it not look "mechanical like."

You can always **add realism to the movements** by animating **secondary motion**. You know, the subtle movement of the toes when the foot hits the ground, delaying the movements of the arms and head due to **inertia**, producing the feel of the weight of the body by lowering the torso a little bit with each step, etc.

That is what I did in general, tracing the basic rough movements, and then adding realism with secondary motion.

The last thing I animated was the facial expression. I did it rather in a hurry, and I made some planning mistakes. To tell you the truth, there was absolutely no planning with this animation, and that was not so very good for the workflow.

Well, the thing was that I had just one hour left before I started rendering, and I didn't have the voice audio edited. Therefore, I had to completely improvise his expressions. Fortunately for me, I did quite a good job

by Juan Pablo Bouza

with the timings, so it was not so difficult to add the voice of the Dwarf over the rendered video. But it would have been much easier and tidier, if I had had the audio track already edited before I animated the character.

This is just a thought though. Luckily enough this movie was rather simple, but if you are going to do a more complex film, it is definitely advisable to have a **rough of the soundtrack already edited before you start animating.**

## Scene Setup

When I first thought about the dancing dwarf scene, I wanted it to have a simplistic background, but simply black was not enough. So, I tried to give some atmosphere to it, and with the use of a **Halo** light I was able to create a rather abstract place that could easily be the dwarf's cave or dungeon. Nothing too specific.

So, this was when the problems began, and I started this technically tricky scene setup... First of all, I used **Blender 2.43/44** for this project, so there were some rendering and lightings features that were not available yet, not even through SVN.

Basically, the scene had two objects: the Dwarf and a simple plane for the ground. What I first did was to add a **Spot Light** with **Halo** enabled. My main concern was the quality of the light and especially of the shadows. Therefore I could not simply use the Spot Light, as spot lights didn't have soft shadows. Besides, I wanted the Volumetric Halo to have **Volumetric Shadows**, and the

only way of doing that was to disable Raytraced Shadows and enable **Buffer Shadows** to the Spot Light.

I was not satisfied with the quality of the shadows that the Spot Light was giving to the scene, so I decided to do a little experiment on **mixing** the **Spot Light** with an **Area Light (which has soft raytraced shadows).**

To make the concept clear, here's how I set it up:

## Dwarf lighting:

- **Spot Dwarf** is the main Halo light, and produces the volumetric shadows inside of the Halo.
- **Area Dwarf** is aligned to Spot Dwarf, and it lit's the Dwarf Mesh and generates raytraced soft shadows over his body.
- **The Dwarf Mesh** is exclusively lighted by the light group called Area, to which Area Dwarf belongs. Spot Dwarf is not part of this group, thus, it does not lit the Dwarf, it just produces the volumetric shadows in the Halo.

## Floor Lighting:

- **The floor mesh** is exclusively illuminated by the lamps that belong to the group called Floor.
- **Spot Floor** is aligned to Spot Dwarf, it has no shadows enabled. Its purpose is to generate the light circle on the floor.
- **Area Floor** is also aligned to Spot Dwarf, and it has the **"Only Shadows"** option enabled, thus, it does not lit anything, but it generates the projection of the raytraced soft shadows of the Dwarf on the floor.



That was the basic lighting setup. But a BIG problem arose... The hair, which had **ZTransp** option enabled, rendered over the Halo light... too bad.

After trying to solve the problem using the Compositor without being able to achieve the desired results, I found out that if you specifically tell the material that has **ZTransp** enabled to be lightened exclusively by a certain group of lights, the **ZTransp** is rendered behind the Halo lights that belong to that group.

So, I had to clone the Dwarf mesh lighting setup and assign those new lights exclusively to the hair. The only problem was that now I had two Halo lights that were summing their Halo values. I solved the problem by lowering each Halo to half of the original value, and that did the trick.

Lastly, there was some strange issue concerning Halo Light rendering against Void and **Ztransp** faces. Some strange white glitches appeared. Fortunately, I was able to solve the problem by **adding a simple plane mesh to the back of the scene**. Therefore, Ztransp and Halo ceased to be rendered against the background black

color, and were rendered against actual black geometry. Well, don't ask me why, but it worked.

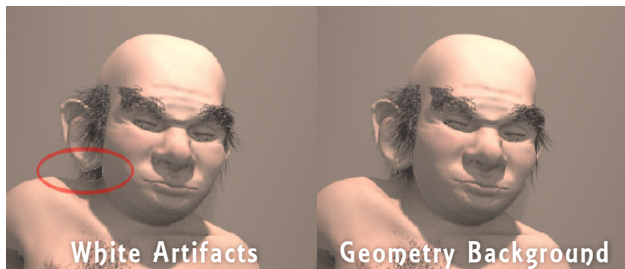
It's amazing how you can solve things by coming up with strange tricky solutions when you are in a hurry... As I said before, I did this animation mainly with Blender 2.43, so many of the technical problems I encountered were fortunately solved in later revisions of Blender.

Well, that's it. I hope this little article is useful for you and encourages you to keep on working on your projects! You can Check out "The Dance of the Bashful Dwarf" animation at [www.blender.org](http://www.blender.org), or at [www.jpbouza.com.ar](http://www.jpbouza.com.ar). Don't forget to check out my new BlenRig and Cartoon BlenRig 1.2 if you enter my site, or at [BlenderArtists](http://BlenderArtists).

I've made the .blend file of the scene with the lighting setup and the BlenRig Armature available to BlenderArt Magazine, so check it out if you want!!

Note: the animation is not compatible with Blender SVN, as it is based on BlenRig 1.0. Try to open it with Blender 2.45 or prior. •

Cheers!  
Juan Pablo Bouza





## Introduction

I first discovered Blender about 6 months ago while I was searching the Internet for animation programmes. It was almost by luck that I found Blender.

I wanted to create some sort of animation and Blender seemed like it could be the right sort of programme to practice on. After searching the Wiki I found out that Blender had a game engine. I then followed a tuto-

rial and created my first Blender game. It was a simple maze and ball game. This taught me about logic bricks and physics.

I then went on trying to create a more complex game. This game was a prison escape game. I learned about bones, alpha textures and fading scenes. My next game was an FPS (first person shooter). I learned how to create a first person game with mouse-camera control. Once I had learned what I wanted, I started the project I am currently working on.

Up until now, all of my learning has come from trial and error and fabulous feedback from [www.Blenderartists.org](http://www.Blenderartists.org). When I first started my current project, I intended it as a learning experience about Blender and its game engine. As people learned about my game and commented, it became more involved.

During this project I have learned about shadow baking, properties and more about modeling. I intend to continue on with this game for some time while I continue to learn more about the power of Blender.

Here are some screen shots of my game. It will be something like the Elder Scrolls series.

Cheers,  
Rusty246 (12 year old student)

Video of [walkthrough](#) of part of the game area: Forum thread on [www.blenderartists.org](http://www.blenderartists.org) showing progress of his game. •





## Introduction

The idea was to create something playful and interactive for my 8-year-old daughter, apart from all that gloomy animation stuff I usually work on. It ought to be simple and funny. I had a look at Blender's game engine and was amazed by its physics engine, "Bullet."

So my daughter and I thought about a character that could be shot and

would then blow up into a big balloon that keeps rolling around. First I began with Blender's "Suzanne" that turned into a donut when it was hit. We discovered that the fun increased with every new bouncing donut.



After a few weeks, when the logic of the game was nearly finished, we came upon the idea of changing the character to a dwarf. So the character, environment, and look of the game were the last things in the game's small production line.

While all of the computer work was done by me, my daughter helped me with the textures, nearly all of which were hand-drawn on paper. Some friends of mine are working on noises and sounds that will be added when they are finished.



I have tried to keep everything as simple as possible, from the installer to playing the game to uninstalling it.

The installer/uninstaller was made with NSIS, a GPL installing system. The game has a splash screen with all necessary instructions. Dwarf Hunt is a fine game for a coffee break, especially if you are angry with somebody. You will only need one hand and a three-button mouse.

Note that the middle mouse-button is the "boss comes around the corner (quit game)" button. Dwarf Hunt can be downloaded at [www.marc-brink.com](http://www.marc-brink.com).

Marc



by Marc

# MEET THE STAFF

54

BlenderArt Staff - No Particular Order

## Joshua Scotton

Hi my name is Joshua Scotton and I'm from England. I'm 21 years old and have just started my fourth year doing an honours degree in Mathematical and Computer Studies.



In my spare time I like to do paint balling, rock-climbing, programming and blender (which I run on a laptop running Ubuntu).

I also run the newly started Blender Week Day Challenge over on [blenderartists.org](http://blenderartists.org). You can find me there under the alias blackcougar.

[www.joshuascotton.com](http://www.joshuascotton.com)

## Kevin Braun

Kevin Braun (HumanForum) is a multimedia professional specializing in all forms of interactive development. He received his BFA from the University of Massachusetts at Dartmouth in 1995 and since then has served world-class clients such as Harvard University, MIT and Cisco Systems.

When he and his wife Christine take a moment from raising their 3 boys they conspire on how to save the world using their combined artistic skills.

To learn more or see examples of his work visit:

[www.brauninteractive.com](http://www.brauninteractive.com)

## Ronan Posnic

After a few years studying philosophy and then communication, in France and Ireland, I finally find myself working in Paris for a small production company.

I gradually improve my skills in Blender so as to integrate it more and more into my own work flow.

Contributing as a proofer is a way for me to give back a little and to follow more closely BlenderArt Magazine issues...

# MEET THE STAFF

55

## Mark Warren

I have been a software developer for 17 years and have finally started to try my hand at CG.

I have always been fascinated with the world of computer graphics although up until now I have only dabbled in trying to produce my own images and animations, but with Blender I hope to start producing some work I can be proud of.

I am currently trying to write a 3D game and I intend to use Blender to do the modeling.

## Scott Hill

Scott Hill began animating professionally in 1992 and over the years has worked extensively in broadcast and film from his home base near Milwaukee, Wisconsin (USA).

Scott enjoys artistic challenge and keeping up with the ever changing landscape of the industry.



## Wade Bick

Wade wants to be a graphic artist when he grows up. However, Wade's already 37 and has a full time job as a systems engineer with a technology company. He also has a wife and 3 kids to support so changing careers at this point is a little risky. Really mastering Blender might have to wait for retirement. In the meantime, he tries to keep up with the rapid advances in Blender by reading BlenderNation, BlenderArt Magazine, and whatever other Blender stuff he has time for (remember the part about the full time job and 3 kids?)



Actually, he can't wait until his kids are a little older and producing 3D-animated shorts can be a fun family pastime. And if anybody in the central Pennsylvania area is interested in meeting to talk about Blender sometime, drop him an email.



# MEET THE STAFF

56

BlenderArt Staff

## Brian Treacy

Aged 29,  
Chesapeake,  
Va

### Occupation:

Senior CADD  
Designer - PB Americas, Inc. in  
Norfolk, Va

**Family Life:** Married Vallari  
January 30, 2002; Lydia born  
November 23, 2006

**Religion:** I enjoy teaching a  
Sunday School class to a group  
of 4th, 5th and 6th graders at  
Freedom Baptist Church.

**Hobbies:** Reading, art, 3D, pro-  
gramming, writing, teaching,  
camping Blender Experience:  
Blender user since April 2002  
as a hobby; first professional  
project in Blender in 2005.

**Writing Experience:** Authored  
many CADD related articles  
published within company's  
internal magazines; externally  
published two articles (one of  
which includes Blender con-  
tent) and one ad; currently  
working on a short book: a  
commentary on the first epis-  
tle of John



## Eric Pranausk

I have been drawing and  
sculpting for as long as I can  
remember, with only minimal  
academic training. I found that  
my creative skills easily car-  
ried over to digital media,  
which I found to be the most  
freeing way to express ideas.  
Discovering open-source soft-  
ware and Blender in particular  
has further helped me to bring  
my imagination to life, not to  
mention put food on the table!

After years of personal experi-  
mentation, guidance from  
members of various online  
communities and employment  
as a graphic designer, I hope  
to help others enjoy a similar  
experience.

## Phillip Ryals

Phillip  
Ryals  
(paryl)  
has been  
around  
the



Blender community since  
1999. He eats and drinks 3D,  
but he finds most of his time  
taken up by the 'real world'.

Until hobbies start paying  
bills, he's more than happy to  
sit in the shadows, quietly  
proofing articles for the best  
Blender magazine out there.

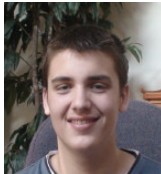
## Noah Summers

(aka BlendRoid).

Drawing, movie making, and miniature modeling have been favorite pastimes of mine. About a year ago, 3d modeling and animation were added to the list, the best part is that all the aforementioned activities relate together.

After trying programs like Anim8r, I was awestruck when I came across Blender.org. I scanned the website multiple times for any diabolical fine print or 40 day trial tags, and was elated to find nothing of the kind!

I've been using Blender obsessively ever since, and can honestly say that it's one of the greatest programs ever to grace the earth with its presence free of charge. ;) I had always planned to have a career in art and computer graphics, but Blender opened a wide new range of opportunities I wouldn't have dreamed of before!



## Derek Marsh

Hello Blender world. My name is Derek Marsh, (BgDM on the forums). Most of you read about me in the last "Meet the Blenderhead" article, so I am not going to go on and on here.

I am from Canada, just outside Toronto. Regular daily life is taken up by my job as a sales manager for [Kinetics Noise Control](#).

If anyone is interested, we are manufacturers of noise and vibration control products. Interesting stuff. I wouldn't change it for anything.

The rest of my time is taken up spending time with my family. Married to a fantastic woman and we have 3 kids. So life is very busy.

Happy Blending!



## Henriël Veldtmann

Henriël is pretty new to Blender, having been using it for only about six months now. Formerly an IT graduate, he started studying Multimedia and Production, where he was introduced to Maya.

After realizing that he'd never be able to afford Maya, he started looking into Blender as an alternative.

And after a short while of struggling with the interface, he decided that Blender is not only the coolest 3D package out there, but just in general the coolest program he's ever seen.

# MEET THE STAFF

58

## BlenderArt Staff

### Joshua Leung

Joshua Leung is an active developer of Blender, with a primary focus on the Animation Tools (in particular the Action Editor, Armatures, and Constraints).

He started using Blender shortly after the release of 2.33a, having previously played around with other freely available 3D-software including Anim8or and POV-RAY.

Currently he is busy working on improving the Animation System to satisfy the requirements of the Peach Open Movie Project.

Also on our staff:

**Bruce Westfall**

**Lynda**

**Patrick O'Donnell**

**Gaurav Nawani**

**Sandra Gilbert**

**Nam Pham**

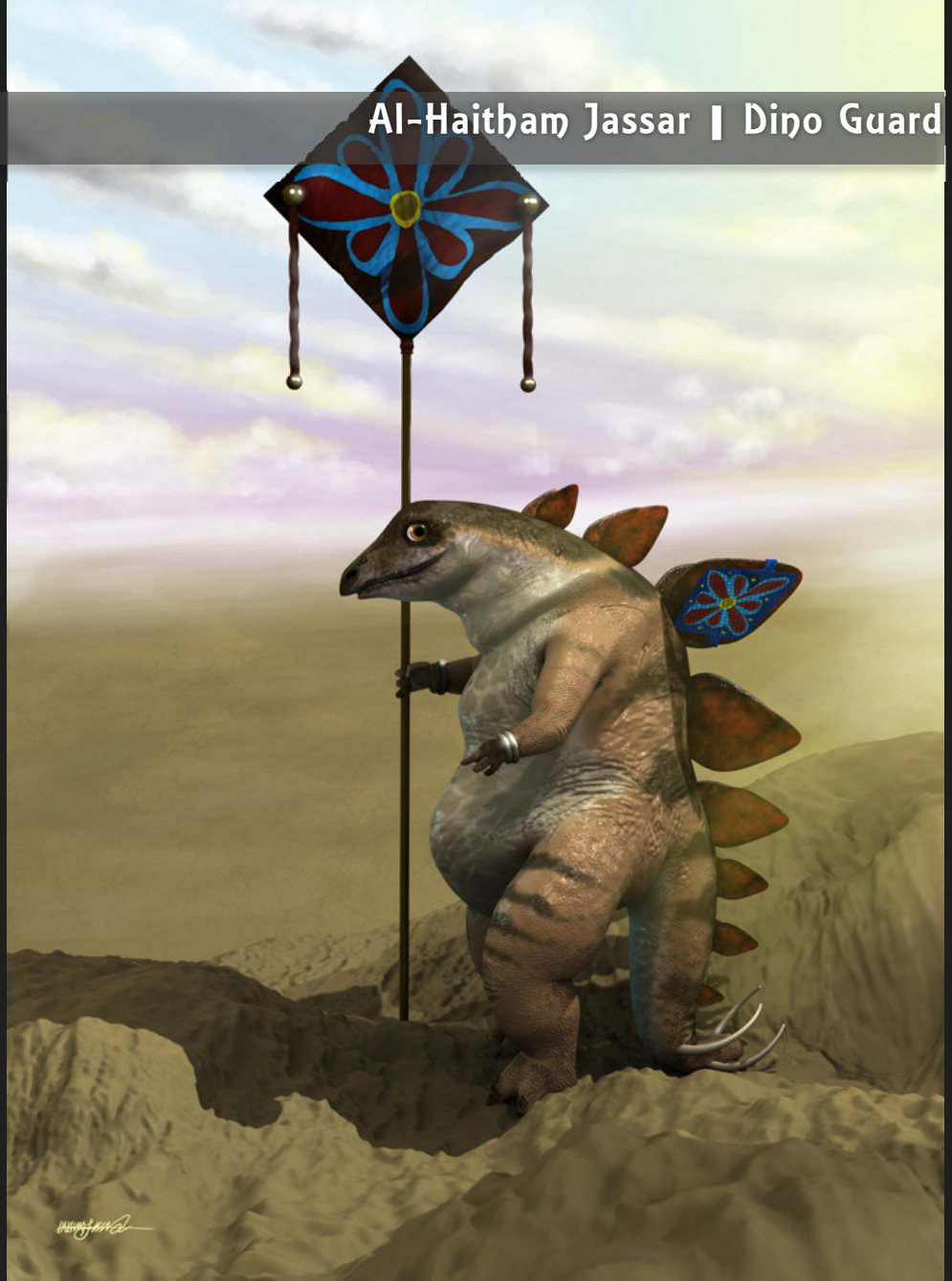






(c) Cristian Mihaescu (criss) 2007





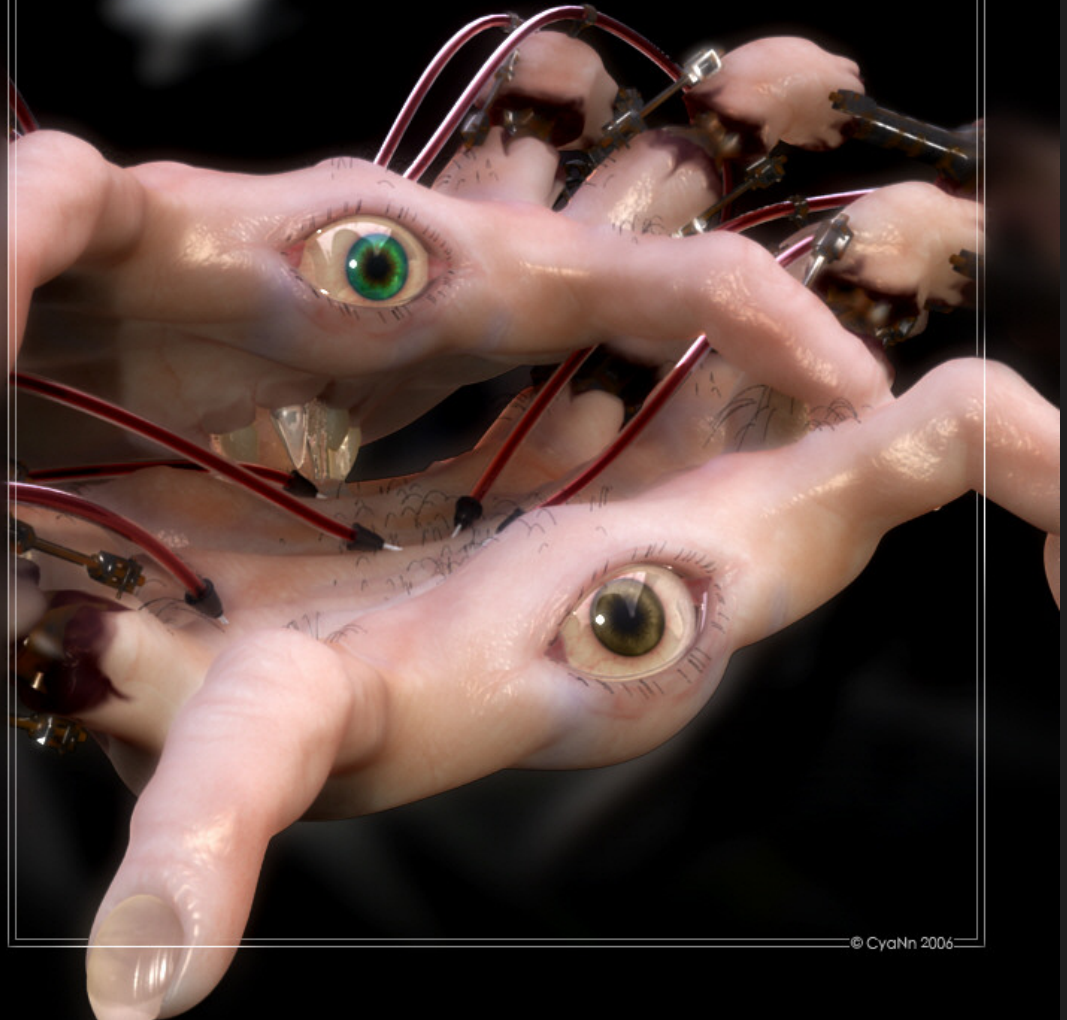




MANHUNT  
Steven Keefer 07







© CyaNn 2006



















# Want to write for BlenderArt Magazine?

72

## Here is how!

### 1. We accept the following:

- Tutorials explaining new Blender features, 3dconcepts, techniques or articles based on current theme of the magazine.
- Reports on useful Blender events throughout the world.
- Cartoons related to blender world.

### 2. Send submissions to [sandra@blenderart.org](mailto:sandra@blenderart.org). Send us a notification on what you want to write and we can follow up from there. (Some guidelines you must follow)

- Images are preferred in PNG but good quality JPG can also do. Images should be separate from the text document.
- Make sure that screenshots are clear and readable and the renders should be at least 800px, but not more than 1600px at maximum.
- Sequential naming of images like, image 001.png... etc.
- Text should be in either ODT, DOC, TXT or HTML.
- Archive them using 7zip or RAR or less preferably zip.

### 3. Please include the following in your email:

- Name: This can be your full name or blenderartist avatar.
- Photograph: As PNG and maximum width of 256Px. (Only if submitting the article for the first time )
- About yourself: Max 25 words .
- Website: (optional)

Note: All the approved submissions can be placed in the final issue or subsequent issue if deemed fit. All submissions will be cropped/modified if necessary. For more details see the blenderart website.

## Issue 14

### Cartoon Modeling/Game Engine

- Adorable/odd characters.
- Scenes in realistic or Toon shading.
- Any game you are working on creating?
- Low poly models.
- Levels.
- How to's on game AI/game play.

## Disclaimer

blenderart.org does not takes any responsibility both expressed or implied for the material and its nature or accuracy of the information which is published in this PDF magazine. All the materials presented in this PDF magazine have been produced with the expressed permission of their respective authors/owners. blenderart.org and the contributors disclaim all warranties, expressed or implied, including, but not limited to implied warranties of merchantability or fitness for a particular purpose. All images and materials present in this document are printed/re-printed with expressed permission from the authors/owners.

This PDF magazine is archived and available from the blenderart.org website. The blenderart magazine is made available under Creative Commons' Attribution-NoDerivs 2.5' license.

### **COPYRIGHT© 2007**

'BlenderArt Magazine', 'blenderart' and BlenderArt logo are copyright of Gaurav Nawani. 'Izzy' and 'Izzy logo' are copyright Sandra Gilbert. All products and company names featured in the publication are trademark or registered trade marks of their respective owners.